

Trabajo Fin de Grado

Desarrollo de una aplicación para el control remoto de cargadores de coches eléctricos a través de Servicios Web

Autor: Daniel Salvador Urgel
Director: Aitor Alcrudo Sangrós
Ponente: Francisco J. Fabra Caro

Grado en Ingeniería Informática
Zaragoza, julio 2016



Universidad
Zaragoza



Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza



DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./D^a. Daniel Salvador Urgel,

con nº de DNI 25207976E en aplicación de lo dispuesto en el art.

14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster)
Grado en Ingeniería Informática, (Título del Trabajo)
Desarrollo de una aplicación para el control remoto de cargadores de coches
eléctricos a través de Servicios Web

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, a 23 de junio de 2016

Fdo: Daniel Salvador Urgel

Desarrollo de una aplicación para el control remoto de cargadores de coches eléctricos a través de Servicios Web

Resumen

Durante los últimos años se ha producido un considerable aumento de las presentaciones hardware en dispositivos móviles que ha generado un cambio de paradigma en el uso de los mismos. Éstos han dejado de ser meros terminales de comunicación para pasar a estar integrados en todos los aspectos de nuestra vida. En este contexto, todos los sectores están adaptando su forma de operar, así como de generar y administrar su información, para poder aprovechar las oportunidades de alcance que ofrecen los Smartphone.

Este Trabajo Fin de Grado pretende aprovechar los avances en la tecnología móvil para crear una aplicación que permita monitorizar la recarga del vehículo eléctrico de forma remota, en tiempo real y para cualquier tipo de cargador, ofreciendo servicios adicionales como la gestión de reservas de los cargadores o la consulta de estadísticas de consumo. Gracias a este sistema se evita tener que llevar siempre consigo una tarjeta RFID para autenticarse y se pasa a utilizar únicamente el teléfono móvil para gestionar las operaciones, lo que reemplaza los actuales PanelPCs, de un elevado coste económico. Todo con el objetivo final de mejorar la experiencia de usuario en los sistemas de recarga de vehículo eléctrico.

La aplicación ha sido desarrollada en Android nativo y se comunica a través de servicios web basados en SOAP con el Centro de Control, que es la aplicación web encargada de la gestión y mantenimiento de los puntos de recarga. Para obtener la información de la estación en la que se quiere recargar se ha diseñado una codificación en formato QR que identifica cada una de las tomas o conectores. Una vez se ha escaneado el código, el usuario puede iniciar el proceso de recarga desde su dispositivo móvil y ver el progreso, así como el resumen de la carga una vez haya finalizado. Por otro lado, se ha asegurado la comunicación a través del módulo de seguridad Rampart, que proporciona implementaciones del protocolo WS-Security para servicios web desarrollados con Axis2.

Índice de contenidos

Capítulo 1	1
Introducción	1
1. Motivación y objetivo final.....	2
2. Marco metodológico	3
3. Marco tecnológico	4
4. Terminología.....	5
5. Organización de la memoria.....	6
Capítulo 2	7
Análisis del problema y requisitos	7
1. Visión general	7
2. Estado del arte	8
3. Análisis de requisitos	9
3.1. Requisitos funcionales.....	9
3.2. Requisitos no funcionales.....	10
4. Plataforma de desarrollo	11
Capítulo 3	12
Arquitectura del sistema	12
1. Esquema global.....	12
2. Comunicaciones	13
Capítulo 4	14
Desarrollo de los servicios web	14
1. Tecnología empleada	14
2. Funciones y datos complejos	14
Capítulo 5	17
Desarrollo de la aplicación Android	17
1. Esquema y organización	17
2. Funcionalidades y secciones de la aplicación.....	18
2.1. Login	18
2.2. Información personal, ajustes y atención al cliente.....	19
2.3. Gestión de cargas	19
2.3.1. Código QR e información del cargador	19
2.3.2. Datos de carga remota	20
2.3.3. Finalización del proceso	20
2.4. Reserva de cargadores	21

2.5. Consulta de estadísticas	22
2.6. Mapa de cargadores.....	23
3. Persistencia	24
4. API móvil de los servicios web	25
Capítulo 6	26
Implementación de la seguridad de los servicios web	26
1. Username Token	26
2. Asegurando la comunicación	28
2.1. Alternativa inicial: seguridad a nivel de mensaje.....	28
2.2. Implementación final: cifrado de la contraseña y seguridad a nivel de capa de transporte .	29
3. La configuración de la seguridad en Android	30
Capítulo 7	31
Pruebas y tests.....	31
1. Pruebas unitarias.....	31
2. Pruebas de integración.....	31
3. Test de usabilidad	32
3.1. Resultados.....	32
Capítulo 8	34
Conclusiones y líneas futuras	34
1. Resultados finales	34
2. Conclusión personal	35
3. Líneas futuras	36
Capítulo 9	38
Bibliografía.....	38
Anexo A	40
Gestión del proyecto.....	40
Anexo B	42
Análisis y diagramas del trabajo.....	42
Anexo C	50
Configuración de WS-Security	50
Anexo D	55
Ciclo de vida de la aplicación	55
Anexo E	56
Descripción y resultados de las pruebas y test	56

Índice de figuras

Figura 1. Distribución de versiones de Android.....	11
Figura 2. Arquitectura del sistema.....	12
Figura 3. Aplicación: imagen principal y menú lateral.	17
Figura 4. Aplicación: login.	18
Figura 5. Aplicación: información del cargador.	19
Figura 6. Aplicación: monitorización del proceso de recarga.....	21
Figura 7. Aplicación: reservas.	22
Figura 8. Aplicación: estadísticas y cargas realizadas.....	22
Figura 9. Aplicación: mapa de cargadores.....	23
Figura 10. Username token con formato PasswordDigest.....	26
Figura 11. Mensaje SOAP.....	27
Figura 12. Algoritmo de doble cifrado con WS-Security.	28
Figura 13. Configuración SSL de Tomcat.	29
Figura 14. Username Token con formato PasswordText.....	30
Figura 15. Tabla-resumen del test de usabilidad.	33
Figura 16. Diagrama de distribución de trabajo.	41
Figura 17. Primer prototipo en papel.	42
Figura 18. Diagrama de casos de uso.	43
Figura 19. Diagrama de secuencia: nueva reserva.	44
Figura 20. Diagrama de secuencia: monitorizar recargas.	45
Figura 21. Diagrama de secuencia: login.....	46
Figura 22. Mapa de navegación: monitorizar una recarga.....	47
Figura 23. Mapa de navegación: login y menú principal.....	48
Figura 24. Mapa de navegación: mostrar cargadores cercanos.....	49
Figura 25. Mapa de navegación: nueva reserva.	49
Figura 26. Política del algoritmo de doble cifrado (I).....	50
Figura 27. Política del algoritmo de doble cifrado (II).....	51
Figura 28. Ejemplo de mensaje SOAP (II).	52
Figura 29. Política de la seguridad a nivel de transporte.	53
Figura 30. Ejemplo mensaje SOAP (II).	54
Figura 31. Ciclo de vida de la aplicación.....	55
Figura 32. Tabla de test unitarios.....	57
Figura 33. Test de integración.....	58

Capítulo 1

Introducción

Durante los últimos años se ha producido un considerable aumento de las presentaciones hardware en dispositivos móviles que ha generado un cambio de paradigma en el uso de los mismos. Desde la banca online y sistemas de pago pasando por aplicaciones de fitness que miden el número de calorías consumidas, los Smartphone han dejado de ser meros terminales de comunicación para pasar a estar integrados en todos los aspectos de nuestra vida. En este contexto, todos los sectores están adaptando su forma de operar, así como de generar y administrar su información, para poder aprovechar las oportunidades de alcance que ofrecen los Smartphone. Esto es especialmente importante para las tecnologías y negocios emergentes que requieren una interacción aún más cercana con el cliente para ser exitosos en su despegue.

Además, la dependencia actual de la sociedad hacia la tecnología móvil, ha provocado que la industria del desarrollo de Software haya puesto el punto de mira y aumentado la inversión en el diseño e implementación de aplicaciones móviles. Con el fin de mejorar la experiencia de uso, se evalúan con exhaustividad las necesidades del usuario móvil en distintos contextos y situaciones, haciendo verdadero hincapié en la personalización de las aplicaciones, lo que sitúa al cliente por encima de la tecnología y lo convierte en parte importante del proceso.

Esta revolución tecnológica también ha llegado al mundo de la automoción. Grandes marcas como Apple, Google o Microsoft utilizan cada vez formas y protocolos de conectividad del vehículo más sofisticados para ofrecer al usuario los distintos servicios que ya se pueden disfrutar en los dispositivos móviles a través de la pantalla táctil, los botones del volante o, con el fin de limitar las distracciones a la hora de conducir, apostando por el control de voz. Estas innovaciones tienen como objetivo que el centro multimedia del automóvil sea una prolongación más de nuestros Smartphones y convertir al propio vehículo en un copiloto inteligente capaz de ofrecer una nueva experiencia de conducción.

1. Motivación y objetivo final

La idea de la realización de este Trabajo Fin de Grado nace en el Centro de Investigación de Recursos y Consumos Energéticos (CIRCE), más concretamente en el departamento de Integración de Energías Renovables (IER). Uno de los campos de investigación de esta fundación son los cargadores de coche eléctrico, donde encontramos el proyecto SIRVE [1]. El objetivo del proyecto es la instalación de estaciones que faciliten la recarga del vehículo eléctrico y reduzcan su impacto sobre la red, así como la energía consumida.

Cada punto de recarga cuenta con un controlador o Smartbox que realiza la comunicación con el usuario físicamente, a través de PanelPCs, y con el servidor. Estas Smartbox se comunican utilizando OCPP, un protocolo estandarizado que utilizan todos los cargadores de vehículo eléctrico. En cuanto a la autenticación de los usuarios en las estaciones, es necesario disponer de una tarjeta RFID que se valida contra la Base de Datos correspondiente.

Actualmente, todos los cargadores se gestionan a través de una aplicación Web (Centro de Control) que, dependiendo del rol de acceso, permite realizar tareas corporativas, como consultar las estadísticas generales de consumo y actividad de las estaciones de una empresa dada, de mantenimiento, permitiendo a los operarios monitorizar el estado y actividad de los cargadores y, por último, tareas a nivel de usuario, donde los clientes pueden consultar sus estadísticas personales y realizar reservas.

Estos cargadores permiten tres tipos de carga que influyen directamente en la duración [2]. Dependiendo del vehículo y la capacidad de su batería, el tiempo de recarga puede ir aproximadamente desde los quince minutos en modo rápido hasta las ocho horas si se ha elegido el lento. Con el sistema actual es imposible monitorizar el proceso y acceder a los servicios del cargador remotamente, por lo que puede llegar a ser incómodo tener que esperar junto al vehículo las horas que se prolongue la carga, o alejarse de éste y que ocurra algún fallo inesperado.

El objetivo es aprovechar los avances en la tecnología de dispositivos móviles para crear una aplicación que permita gestionar la recarga del vehículo eléctrico de forma remota en tiempo real y para cualquier tipo de estación, ofreciendo servicios adicionales como la gestión de reservas o la consulta de estadísticas de consumo. Además, se evitaría tener que llevar siempre consigo la tarjeta RFID, utilizando únicamente el teléfono móvil para gestionar las operaciones y reemplazando los actuales PanelPCs, de un elevado coste económico. Todo con el objetivo final de mejorar la experiencia de usuario en los sistemas de recarga de vehículo eléctrico.

2.Marco metodológico

Para el desarrollo de este trabajo se ha seguido una metodología ágil que nos ha permitido ir alcanzando los objetivos intermedios propuestos y, de esta manera, poder ir mostrando el resultado del trabajo realizado progresivamente. Frente a un desarrollo tradicional en cascada donde cada etapa del desarrollo del software se ejecuta una única vez, la metodología ágil permite solapar las etapas y volver atrás para redefinir los requisitos y hacer cambios mientras el proyecto avanza, consiguiendo un ciclo de vida iterativo e incremental.

Esto se planteaba necesario debido a que solamente había un requisito inamovible en el proyecto, la monitorización de recargas, y a partir de éste las funcionalidades podían cambiar, aumentarse, o redefinirse en cada iteración para conseguir satisfacer las nuevas necesidades o problemas encontrados. Además, el trabajo se ha realizado de manera individual y, a pesar de estar relacionado con otros proyectos y en contacto con otras arquitecturas, se ha contemplado como un desarrollo aparte por lo que no era necesario tener una documentación firme que fuera aceptada por un grupo de trabajo.

Por todo esto, no se ha decidido contemplar un modelo en cascada, que obliga a tener los requisitos fijos desde un principio y se ha apostado por esta metodología. En un primer momento se establecieron iteraciones de dos semanas pero normalmente se ampliaban a tres o cuatro, en función de la carga de trabajo del objetivo a conseguir.

3.Marco tecnológico

A continuación se muestran los distintos entornos, herramientas y programas utilizados en la realización de este Trabajo Fin de Grado:

- **Eclipse IDE:** Entorno de programación para el desarrollo del servidor y los servicios web.
- **Java:** Lenguaje de programación utilizado para el servidor, los servicios web y complementario a Android.
- **Apache Tomcat:** Contenedor de servlets que aloja la aplicación y servicios web del sistema.
- **Apache Axis2:** Motor de servicios web utilizado para generar los servicios que consumirá la aplicación.
- **SOAP:** Protocolo de comunicación utilizado entre el servicio y la aplicación móvil, que se basa en el intercambio de objetos XML.
- **JAX-WS:** API Java para la construcción de servicios web con Axis2.
- **WS-Security:** Protocolo de comunicaciones utilizado para implementar seguridad en los servicios web. Permite dotar de integridad y confidencialidad a los mensajes SOAP así como incluir tokens de seguridad.
- **Apache Rampart:** Modulo encargado de implementar el estándar de WS-Security en el motor de servicios web Axis2.
- **Wireshark:** Se utiliza para capturar los paquetes comunicados entre la aplicación y el servidor y analizar su contenido.
- **Android Studio:** Entorno de desarrollo de la aplicación.
- **Gradle:** Sistema de automatización de compilación utilizado con Android Studio.
- **Android SDK:** Kit de desarrollo de Software para realizar la programación de Android en nativo.
- **Motorola Moto G:** Teléfono con Sistema Operativo Android 5.1 utilizado para las pruebas de la aplicación.

4. Terminología

A continuación se explican una serie de términos que aparecen en esta memoria relacionados con la gestión de la energía y el vehículo eléctrico:

- **Smartbox:** Sistema empotrado consistente en un panelPC industrial y software desarrollado en Ada que realiza la gestión del cargador de vehículo eléctrico así como la interfaz de usuario.
- **CPMS (Charge Point Management System):** Controlador lógico de un elemento discreto de carga de vehículo eléctrico (p.ej.: Cargador SGTE o Mennekes).
- **Toma:** Conector o manguera de la estación de recarga. En los cargadores utilizados en las pruebas de este trabajo se cuenta con tres tipos: SAEJ112 (carga lenta), Mennekes (carga moderada) y ChaDeMo (carga rápida).
- **Ocpp:** Protocolo de comunicación que utilizan las estaciones de recarga eléctrica.
- **PanelPC:** Sistema de control industrial con pantalla integrada. Dispositivo hardware sobre el que se despliega la lógica de la Smartbox.
- **Tarjeta RFID:** Sistema de almacenamiento y recuperación de datos remoto que transmite la identidad de los usuarios mediante radiofrecuencia. Se utiliza para autorizar a los usuarios en los puntos de recarga.
- **Código QR (Código de respuesta rápida):** Sistema que almacena información en una matriz de puntos y que puede ser leído mediante un dispositivo móvil con cámara. Los tres grandes cuadrados en las esquinas permiten al lector detectar la posición del código y una de sus ventajas es que la lectura se realiza de manera instantánea.

5. Organización de la memoria

La memoria está organizada de la siguiente forma:

- En el **Capítulo 1** se presenta el Trabajo Fin de Grado, la motivación que ha llevado a realizarlo y las herramientas utilizadas para su desarrollo.
- En el **Capítulo 2** se analiza el problema y se definen los requisitos del trabajo y la plataforma de desarrollo.
- En el **Capítulo 3** se define la arquitectura del sistema, así como las comunicaciones necesarias entre los distintos dispositivos.
- En el **Capítulo 4** se describe la implementación de los servicios web y las distintas funcionalidades que ofrece.
- En el **Capítulo 5** se explica el funcionamiento de la aplicación móvil, así como la interacción con los distintos servicios y la gestión de la persistencia.
- En el **Capítulo 6** se detalla la implementación de la seguridad en la comunicación de los servicios web.
- En el **Capítulo 7** se describen las pruebas realizadas para garantizar el comportamiento correcto de la aplicación.
- En el **Capítulo 8** se comentan las conclusiones finales, opinión personal y líneas futuras.
- En el **Capítulo 9** aparece la bibliografía de este trabajo.

Al final de la memoria se adjuntan varios anexos descritos a continuación:

- En el **Anexo A** se ilustra la gestión del proyecto y la distribución de las distintas tareas a lo largo del tiempo.
- En el **Anexo B** aparecen los diagramas correspondientes a la aplicación realizada que describen las relaciones y propiedades, así como el funcionamiento.
- En el **Anexo C** se explica con más detalle la configuración de seguridad así como distintos ejemplos en el intercambio de mensajes.
- En el **Anexo D** aparece un esquema de la gestión del ciclo de vida de la aplicación y su relación con la persistencia.
- En el **Anexo E** se añaden las trazas de las pruebas unitarias y de integración, además la ficha de la prueba de usabilidad y los resultados detallados.

Capítulo 2

Análisis del problema y requisitos

1. Visión general

La finalidad de esta aplicación es mejorar la experiencia de usuario en el proceso de recarga de vehículo eléctrico. Para conseguirlo y atraer el máximo público posible es importante no realizar un simple sistema de carga remota sino desarrollar una aplicación interactiva, sencilla e intuitiva que también englobe otras tareas útiles que se puedan ofrecer al cliente como la consulta de disponibilidad y reserva de cargadores o estadísticas y gráficas de consumo, además de brindar la posibilidad de buscar puntos de carga cercanos. A continuación se describen los puntos fuertes de la aplicación:

- ❖ En el sistema tradicional los usuarios tenían que autenticarse en la estación de recarga mediante una tarjeta RFID. Con la aplicación, la tarjeta podrá dejarse a un lado y realizar la identificación del conductor desde el dispositivo móvil introduciendo el usuario y contraseña de la cuenta personal.
- ❖ Con el objetivo de reemplazar los actuales PanelPC's y gestionar en su totalidad las cargas desde el móvil, se creará un código QR para cada toma de carga. El usuario simplemente tendrá que escanear con la cámara el código deseado y obtendrá toda la información del cargador, así como su disponibilidad y permisos de acceso.
- ❖ Uno de los principales problemas del proceso de recarga es la monitorización de ésta. La solución pasa por recibir actualizaciones periódicas del estado del proceso a través de la aplicación y ser alertado acerca de posibles errores o finalización.
- ❖ Por último, funcionalidades adicionales como la consulta de estadísticas, reserva de cargadores y búsqueda de puntos de recarga cercanos acercan al usuario al sistema y permiten que las opciones disponibles en la aplicación web estén también accesibles desde el dispositivo móvil para disfrutarlas en cualquier momento y/o lugar.

2.Estado del arte

Dado que es una aplicación con un público y plataforma objetivo muy concreto, no se han encontrado aplicaciones que puedan ocupar el mismo sector de mercado y por lo tanto no hay competencia directa. A pesar de ello, si se han hallado otras aplicaciones que incluyen sistemas de localización o reserva de cargadores que pueden servir de inspiración en el desarrollo del trabajo. A continuación se detallan una serie de ellas que han podido aportar ideas ya sea por su cercanía con el tema y funcionalidades de la aplicación o por el diseño innovador:

- eCar: aplicación de Endesa que permite visualizar en el mapa los puntos de carga de coche eléctrico más cercanos, además de planificar la ruta hasta cualquiera de ellos. Por otro lado, te permite la reserva de cargadores y consultar las estadísticas de cargas realizadas con el vehículo en la red de Endesa.

<https://play.google.com/store/apps/details?id=com.enel.sime>

- Ecarga: esta aplicación permite localizar puntos de recarga de vehículo eléctrico en cualquier lugar del mundo, proporcionando información detallada de cada uno de ellos y con la posibilidad de dar feedback o incluir nuevos puntos por parte del usuario.

<https://play.google.com/store/apps/details?id=com.awgs.ecarga>

- Airbnb: esta empresa de alquiler de alojamientos está en plena expansión debido a su nuevo concepto de viajar. Pero no solo eso, el diseño renovado y moderno de su aplicación móvil, además de su alta interactividad con el sistema capta nuestra atención y ofrece una gran experiencia de uso.

<https://play.google.com/store/apps/details?id=com.airbnb.android>

3. Análisis de requisitos

En la sección 2 del anexo B aparece el diagrama de casos de uso de este Trabajo Fin de Grado. A partir de éste y de una manera más formalizada se han especificado los requisitos funcionales y no funcionales que se exponen a continuación ordenados por prioridad de implementación.

3.1. Requisitos funcionales

(Prioridad 1)

- **RF1.** El usuario, al entrar en la aplicación, puede acceder con su usuario y contraseña y ser autenticado por el Centro de Control.
- **RF2.** El usuario puede acceder a su configuración de cuenta y ver sus datos.
- **RF3.** El usuario tiene la posibilidad de ver la información de la toma de un cargador mediante el escaneo de su código QR y ver si está disponible o está reservada/fuera de servicio. Además, el usuario puede ver está autorizado para su uso.
- **RF4.** El usuario puede iniciar una recarga en el cargador que ha escaneado, eligiendo la cuenta (en caso de tener varias) con la que quiere realizar el proceso.
- **RF5.** El usuario puede monitorizar la carga y ver el progreso y estadísticas de la misma.
- **RF6.** El usuario tiene que ser notificado en caso de producirse una parada o fallo en la carga activa.
- **RF7.** El usuario tiene la posibilidad de parar la carga de su vehículo desde la aplicación.

(Prioridad 2)

- **RF8.** El usuario puede realizar una reserva para una fecha y hora dadas en el cargador que seleccione y con la toma que necesita su vehículo.
- **RF9.** El usuario tiene que recibir una confirmación en el caso de que su reserva se haya procesado con éxito o un error en el caso de que no se haya podido completar.
- **RF10.** El usuario puede ver la lista de sus reservas activas.

- **RF11.** El usuario tiene la posibilidad de eliminar una de sus reservas.

(Prioridad 3)

- **RF12.** El usuario puede visualizar las últimas cargas realizadas y un resumen de las mismas.
- **RF13.** El usuario puede ver un gráfico resumen de su consumo energético.

(Prioridad 4)

- **RF14.** El usuario puede visualizar en el mapa los cargadores a los que tiene acceso.
- **RF15.** El usuario, dada su localización GPS, tiene la posibilidad de filtrar los cargadores que tiene disponibles en función de un radio de distancia.

3.2.Requisitos no funcionales

- **RNF1.** El dispositivo contará con el Sistema Operativo Android.
- **RNF2.** La versión Android del dispositivo es igual o mayor que la 4.1 (Jelly Bean API 16).
- **RNF3.** El dispositivo cuenta con cámara.
- **RNF4.** El dispositivo cuenta con conexión a internet.
- **RNF5.** El dispositivo cuenta con GPS.

4. Plataforma de desarrollo

Como sistema operativo se ha optado por el uso de Android dada su elevada audiencia y dominio del mercado (82.8% frente al 13.9% que tiene iOS y el 2.6% de Windows Phone [3]) y la posibilidad de testear más fácilmente debido a la oportunidad de acceder a dispositivos con distintas versiones de este Sistema Operativo.

Android se adapta a los cambios tecnológicos y a las nuevas corrientes en cuanto a diseño se refiere. Por ello, las nuevas versiones introducen guías de estilo más novedosas y que permiten al usuario gozar de una mayor experiencia visual. Es el caso de Material Design, una guía integral para el diseño visual, de movimientos y de interacción, que fue presentada en el Google I/O 2014 para acompañar a la versión 5.0 [4]. Para fijar la versión mínima de la aplicación es necesario decidir entre innovación o compatibilidad ya que las nuevas librerías pueden no ser soportadas por versiones más desfasadas. Para resolver este inconveniente se ha recurrido a las estadísticas oficiales proporcionadas por Android y que se pueden observar en la figura 1. Dado que un 92.7% de los usuarios poseen una versión igual o superior a la 4.1 (Jelly Bean API 16) y, por lo tanto, tan solo un 7.3% siguen por debajo de ella, utilizaremos esta versión como mínima de compatibilidad.

Version	Codename	API	Distribution
2.2	Froyo	8	0.2%
2.3.3 - 2.3.7	Gingerbread	10	3.8%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	3.3%
4.1.x	Jelly Bean	16	11.0%
4.2.x		17	13.9%
4.3		18	4.1%
4.4	KitKat	19	37.8%
5.0	Lollipop	21	15.5%
5.1		22	10.1%
6.0	Marshmallow	23	0.3%

*Data collected during a 7-day period ending on November 2, 2015.
Any versions with less than 0.1% distribution are not shown.*

Figura 1. Distribución de versiones de Android.

Capítulo 3

Arquitectura del sistema

En este capítulo se describe el esquema general del trabajo, cuáles son las partes que lo componen, cómo se comunican entre sí y cuáles han sido las decisiones más importantes que se han tomado en el desarrollo del mismo.

1. Esquema global

El proyecto podría dividirse en cuatro unidades, las cuales están interconectadas a través de distintos protocolos de comunicación y se pueden visualizar en la figura 2. En primer lugar, tenemos las estaciones de recarga de vehículo eléctrico, que son gestionadas a través de Smartbox y se comunican con el Centro de Control, que es la aplicación web encargada del acceso, mantenimiento y monitorización de los cargadores. También es la responsable de la autenticación y autorización de los usuarios contra la Base de Datos relacional gestionada con PostgreSQL. Por último, la aplicación móvil, base de este trabajo, accede a las distintas funcionalidades del Centro de Control, comunicándose con las estaciones a través de él y convirtiéndose en un sustituto portable de la aplicación web.

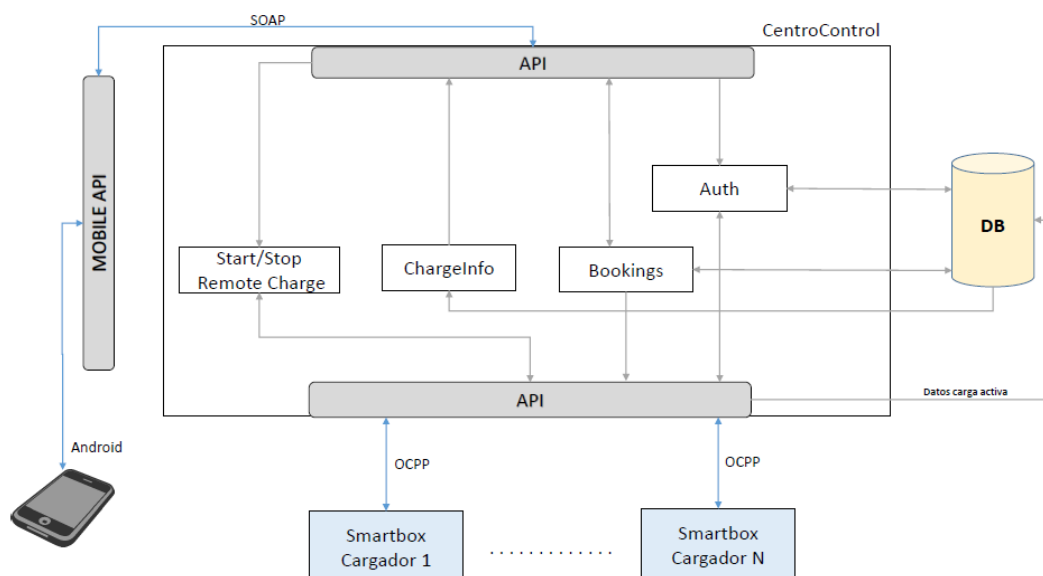


Figura 2. Arquitectura del sistema.

En un primer momento se pensó en realizar la comunicación de la aplicación móvil directamente contra las Smartbox. Esta opción tenía la ventaja de que sería válida para cualquier estación de carga que cumpliera el estándar, pero tenía la desventaja de que el módulo de autenticación y otros servicios necesarios están englobados dentro del Centro de Control. Este hecho provocaría que conectar directamente con los cargadores conllevaría que éstos tuvieran que pasar por la aplicación web en la ejecución de las tareas con la aplicación móvil, provocando un mayor costo en el procesamiento de las operaciones.

Finalmente, se decidió realizar una API en el Centro de Control que permitiera acceder a través de servicios web a las distintas funcionalidades y acciones que están disponibles tales como autenticar a un usuario, iniciar una carga remota, obtener información de los cargadores disponibles o reservar una toma.

2. Comunicaciones

Las estaciones de recarga se comunican con el Centro de Control a través de OCPP (Open Charge Point Protocol), protocolo de envío de paquetes basado en SOAP. Este estándar de comunicación fue propuesto por la *Open Charge Alliance* (OCA), un compendio de empresas dedicadas a los cargadores de vehículo eléctrico, con el fin de ofrecer una solución uniforme para el método de comunicación entre los puntos de recarga y el sistema central [5]. Así, se consigue que cualquier estación pueda conectar con el sistema de control sin tener en cuenta el proveedor.

Para la comunicación entre la API móvil y el Centro de Control se ha elegido SOAP. Este protocolo puede usarse sobre HTTP o HTTPS y está basado en el intercambio de datos XML, que facilita la lectura y el intercambio de información compleja y estructurada. Por contra, también hace que los mensajes sean más pesados y, por lo tanto, más lentos de transferir. Este hecho no supone un gran impedimento ya que no se necesita transmitir un gran volumen de información.

Una de las ventajas más importantes radica en que este protocolo está estandarizado por el *World Wide Web Consortium* y permite establecer seguridad a nivel de mensaje mediante WS-Security. Esta implementación permite asegurar la comunicación *end-to-end*, por lo que la existencia de intermediarios durante el camino del mensaje no altera la confidencialidad e integridad del mensaje. Esto es posible gracias a que se puede elegir qué partes del mensaje van cifradas y para quién en concreto, dotando al protocolo de cierta granularidad para permitir el acceso a determinados usuarios [6]. Por otro lado, la seguridad a nivel de transporte sobre SSL, más ligera al no modificar el contenido de los mensajes, cifra el canal de comunicación en su totalidad y ofrece seguridad *point-to-point*, provocando que un mensaje fuera de ese recorrido pueda ser visualizado por cualquier usuario. La decisión final de implementación, contemplando la posibilidad de ambos sistemas, se expone en el capítulo 6.

Capítulo 4

Desarrollo de los servicios web

En este capítulo se detalla el proceso de creación de los servicios web, qué tecnologías y métodos se han empleado, y cuáles son las funciones y tipos de datos complejos que se han implementado para satisfacer los requisitos de la aplicación.

1. Tecnología empleada

Los servicios web del Centro de Control han sido desarrollados con el motor Apache Axis2 siguiendo el modelo estandarizado y basado en anotaciones JAX-WS. Esta elección se debe a que tiene soporte para Spring, framework utilizado para el desarrollo de la aplicación web y es compatible con SOAP, tanto la versión 1.1 como la 1.2 [7]. Además, está diseñado para soportar la adición de módulos que extienden la funcionalidad para implementar características tales como la seguridad o la fiabilidad. Es el caso de WS-Security, soportado con Apache Rampart y utilizado para hacer seguros los servicios web del proyecto, y WS-Addressing.

Dado que construir manualmente el WSDL puede resultar un proceso complejo, se ha optado por desarrollar una interfaz basada en la funcionalidad que debe tener el servidor Web e implementar los métodos Java de los distintos servicios. A partir de esta clase, en la que están definidas las distintas operaciones, datos y *bindings* con el módulo de Spring, podemos generar el WSDL. Este proceso se conoce como *Bottom-up Web Services* [8].

2. Funciones y datos complejos

El servidor proporciona una serie de métodos a los que se puede acceder a través de servicios web y que permiten interactuar con el Centro de Control y las estaciones de recarga. Para el intercambio de información ha sido necesario recurrir a datos complejos ya que, por ejemplo, para enviar los datos de un usuario, necesitaremos crear un tipo *UserData* con los atributos y la información requerida por el servidor para definir este objeto. A continuación, se detallan todas las estructuras que han sido definidas para los servicios:

- **AccountData:** objeto que define una cuenta de usuario y que está definido por el identificador de ésta y el número de tarjeta asociado.
- **UserData:** contiene toda la información personal del cliente (nombre, apellidos, email, teléfono y DNI), además de la lista de reservas activas, las cuentas que posee y la lista de operaciones o cargas que ha realizado en el sistema.
- **BookingData:** estructura que define la reserva de un cargador. Está formada por la fecha, las horas de inicio y fin, el identificador de la toma y el tipo de carga seleccionado.
- **MeasureData:** objeto que representa una medida de carga intermedia y está formado por toda la información relevante de ésta: tiempo transcurrido, voltaje, corriente, energía y, en el caso de carga rápida, porcentaje de carga y tiempo restante.
- **MeasureDataArray:** estructura que define una recarga completa y que contiene una lista de todas las medidas intermedias recibidas, además de la información acerca de la energía total, duración y tipo de carga.
- **ChargeDataResponse:** define la respuesta del servidor a la consulta de información de un cargador a través de su código QR. Está compuesta por el estado del cargador (disponible, no disponible o reservado) y la autorización o permisos (autorizado-true o no autorizado-false).

Las funciones disponibles en el servidor web son las siguientes:

- **Authenticate:** el usuario accede a la aplicación con su usuario y contraseña. El Centro de Control recibe estos datos y autentica al usuario, devolviendo los datos del cliente en caso satisfactorio o error en el caso de que no coincidieran el usuario y/o la contraseña.
- **GetChargerState:** el usuario lee el código QR, que devuelve los ID's de cargador, cpms y toma. Esta función manda esos parámetros, además del ID del usuario. El Centro de Control comprueba el estado del cargador (por ejemplo, si está reservado, comprueba si el usuario es el que ha hecho la reserva), y devuelve un mensaje con la situación de este (Disponibilidad y autorización).
- **StartCharge:** este método manda una petición de inicio de carga con los parámetros de la toma, CPMS y el número de tarjeta del usuario. Recibe un mensaje con la aceptación o el rechazo del comienzo remoto.

- **StopCharge:** este método manda una petición de finalización de carga con el ID de la toma y el del CPMS donde se está realizando. Recibe un mensaje con la aceptación o el rechazo de la parada remota.
- **GetChargeData:** el método envía como parámetro el ID de la toma y del CPMS donde se está realizando la carga, y el número de medidas que se desea recibir. Devuelve la información relativa a la carga en el caso de que haya una activa en esa toma o null en el caso de que no haya ninguna en el momento de la petición.
- **BookCharger:** envía como parámetros el ID del cargador, el de la toma, y las fechas de inicio y fin de la reserva. El centro de control devuelve “ok” en el caso de haberse completado, “ocupado” en caso de que no esté disponible, o “error fechas”, si el usuario ha introducido una fecha de inicio posterior a la de fin o el tiempo máximo de reserva ha sido superado.
- **DeleteBooking:** elimina la reserva de un usuario indicando por parámetro el ID de ésta.
- **GetUserBookings:** devuelve la lista de reservas para un ID de usuario dado.
- **GetLastOperations:** devuelve la lista de las últimas cargas realizadas por el usuario indicando el ID del cliente y el número de cargas que se quieren obtener.
- **GetAvailableChargers:** devuelve la lista de estaciones de carga disponibles para un ID de usuario dado.
- **GetEnergyYear:** devuelve la lista de consumo energético por mes del año actual indicando por parámetro el ID del usuario.
- **GetUserConsumption:** envía como parámetros el ID del usuario y un rango de fechas, y devuelve el consumo energético del usuario en ese periodo de tiempo.

Capítulo 5

Desarrollo de la aplicación Android

En este capítulo se encuentra toda la información en relación al diseño e implementación de la aplicación móvil, así como una descripción de cada pantalla y las interacciones con los servicios web necesarias para mostrar los datos o llevar a cabo las acciones que se ofrecen.

1. Esquema y organización

La aplicación está dividida en una serie de pantallas o *activities*, que contienen las funcionalidades principales de la aplicación. La *activity* principal (figura 3), a la que se accede tras autenticarse en la aplicación, consiste en un *View Pager* que permite navegar por los distintos menús con un simple gesto de dedo, deslizando hacia derecha o izquierda [9]. Este diseño permite darle a la aplicación un aspecto atractivo y limpio, sin tener que recurrir a botones o enlaces para desplazarnos por los menús. La vista está gestionada por un adaptador que genera las distintas páginas alojadas como fragmentos, y que el usuario visualiza conforme desliza la pantalla o pulsa en las respectivas pestañas.

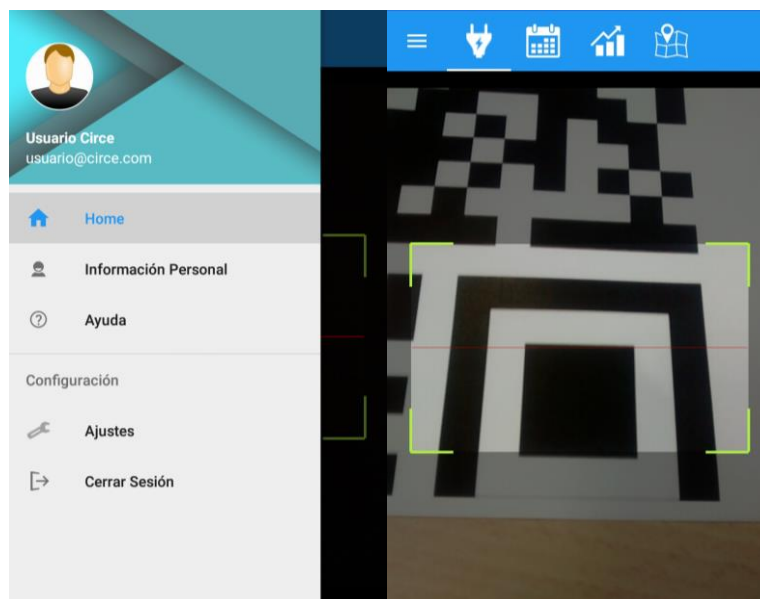


Figura 3. Aplicación: imagen principal y menú lateral.

Por otra parte, se incorpora un *Navigation Drawer* o menú lateral que se encuentra oculto y al que se puede acceder o bien deslizando el dedo desde el borde izquierdo hacia el centro de la pantalla, o pulsando sobre el botón de inicio ubicado en la esquina izquierda de la barra de acción. Este panel contiene los accesos a la información personal, atención al cliente, ajustes del sistema y cierre de sesión. En la sección 4 del anexo B aparecen los mapas de navegación de la aplicación y a continuación, se exponen con más detalle las distintas pantallas y menús de los que consta la aplicación, así como su comportamiento e información que proporcionan.

2. Funcionalidades y secciones de la aplicación

2.1. Login

La pantalla de acceso (figura 4) sigue un diseño sencillo y funcional que permite a los usuarios entrar a la aplicación con su usuario y contraseña, señalando los posibles errores en cuanto al formato o desigualdad de las contraseñas. Mediante la invocación al servicio *authenticate*, el usuario recibe toda la información de su cuenta personal, así como la lista de cargadores, reservas y últimas operaciones. Esta información se guarda en las variables globales de la aplicación para su posterior uso en las distintas funcionalidades que se expondrán a continuación. Además, en el caso de que entre un nuevo cliente, indicará los pasos a seguir para registrarse y solicitar una cuenta.

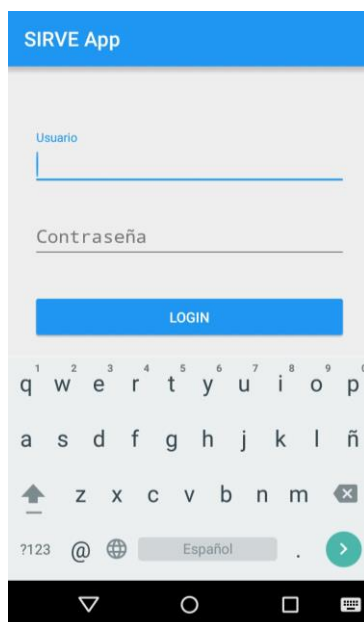


Figura 4. Aplicación: login.

2.2. Información personal, ajustes y atención al cliente

En el panel lateral, el usuario puede consultar la información personal asociada con su cuenta, cambiar los ajustes de la aplicación o saber a quién o dónde dirigirse en el caso de tener cualquier duda o querer notificar algún problema relacionado con el sistema. Además, tiene la posibilidad de cerrar sesión y borrar sus datos de acceso del dispositivo.

2.3. Gestión de cargas

La funcionalidad principal de este trabajo es la del control remoto y la monitorización de recargas de vehículo eléctrico. Nada más acceder a la aplicación, después de autenticarse en el sistema con sus datos personales, el usuario se encuentra un lector de códigos QR que iniciará el proceso de recarga. Las interacciones de este proceso entre los distintos sistemas se encuentran detalladas en forma de diagrama de secuencia en la sección 3 del anexo B.

2.3.1. Código QR e información del cargador

El código QR está diseñado con una codificación especial para evitar que la aplicación analice cualquier tipo de código capturado por la cámara. Una vez el usuario enfoque un código que cumpla esta especificación, se extraen los datos del cargador: nombre del CPMS, ID relativo de la toma, tipo de carga y nombre de la estación de recarga. Posteriormente, se invoca al servicio web *getChargerState*, que indica al usuario si está autorizado a utilizar el punto de recarga solicitado e informa del estado actual del cargador (disponible, ocupado, reservado o fuera de servicio). En este punto, se muestra un botón de inicio de carga (figura 5), que estará activo en el caso de que sean positivas tanto la disponibilidad como los permisos de uso. En caso contrario, cuando el usuario pulsa el botón de inicio, el sistema notifica al usuario de que no es posible iniciar el proceso de recarga.



Figura 5. Aplicación: información del cargador.

2.3.2. Datos de carga remota

En cuanto se pulsa el botón de inicio, la aplicación manda una petición al servicio *startCharge*, que recibirá si se ha aceptado o no el inicio remoto de la recarga en la toma seleccionada. En el caso de que sea aceptada, se manda una serie de peticiones consecutivas al servicio *getChargeData* para comprobar si se ha registrado la carga en el sistema y lleva al usuario a la pantalla de información de carga activa.

Una vez ha dado comienzo la recarga del vehículo, la aplicación lanza en background el ejecutor de tareas que invocará periódicamente al servicio web *getChargeData*. Dado que esta operación solo tiene cabida dentro del ciclo de vida de la aplicación, se usan los recursos de la aplicación en lugar de los del sistema para la planificación de las tareas [10]. Para este cometido, se ha implementado un Handler que invoca a un Thread con la tarea asíncrona cada quince segundos de una manera eficiente y devuelve la energía, en kilovatios-hora, recargada hasta el momento, tal como se puede observar en la figura 6.

2.3.3. Finalización del proceso

La recarga del vehículo puede terminar por orden del sistema, en el caso de una finalización correcta o un error del cargador, o puede ser solicitada por el propio usuario a través del botón disponible en la pantalla de carga activa y que llama al servicio *stopCharge*. Dado que la aplicación tan solo actúa como cliente, no puede recibir mensajes del servidor sin que sean una respuesta a una petición enviada con anterioridad. Esto significa que no se puede informar directamente al usuario de la aplicación de posibles errores o terminación de la recarga. La solución propuesta para este problema se basa en dar por terminada la aplicación cuando se reciba más de una medida nula desde el centro de control, que simboliza que no hay ninguna carga activa en la toma.

Una vez se ha terminado el proceso, la aplicación muestra una pantalla (figura 6) con el resumen general de la carga: duración, energía total suministrada y la información de la estación utilizada.

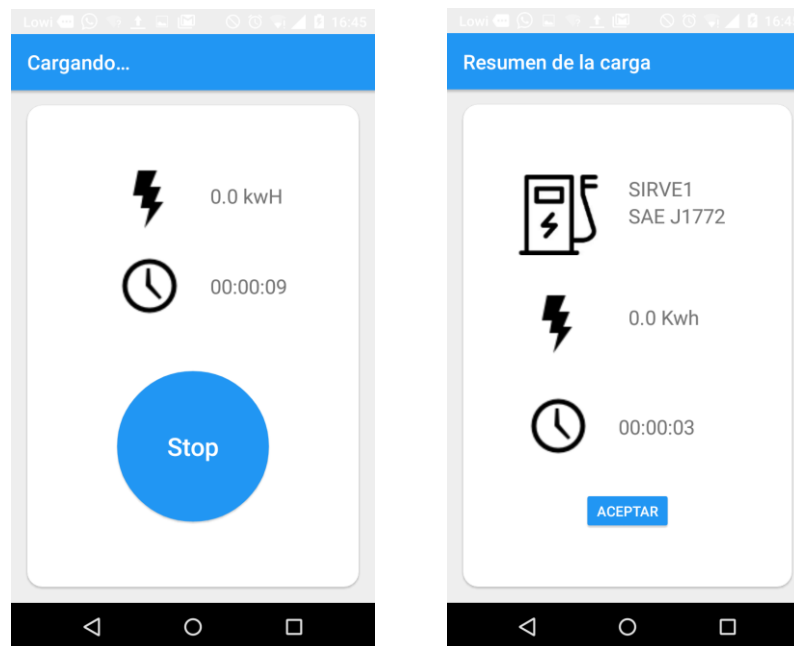


Figura 6. Aplicación: monitorización del proceso de recarga.

2.4. Reserva de cargadores

Otra de las funcionalidades de la aplicación es la reserva de cargadores. En esta pantalla, el usuario puede ver todas sus reservas, tanto activas como pasadas, y puede cancelarlas si lo desea. Por otro lado, es posible agregar una nueva reserva pulsando en el botón flotante, ubicado en la esquina inferior derecha (figura 8).

Para llevar a cabo una nueva reserva es necesario invocar al servicio *bookCharger* especificando la fecha, hora de inicio y fin, el nombre de la estación de recarga, y el tipo de toma. El sistema busca una toma disponible dentro de esos parámetros y notifica el usuario si la reserva ha sido aceptada, el cargador está ocupado o si las horas introducidas son erróneas (máximo número de horas excedido, u hora de inicio posterior a la de final). En caso de que la operación se realice con éxito, se invocará al servicio *getUserBookings* para recargar la lista de reservas del usuario con la nueva información. Esto es necesario debido a que el método para eliminar reservas, *deleteBooking*, necesita el ID único de la operación, generado en el servidor a la hora de registrar la reserva en la base de datos.

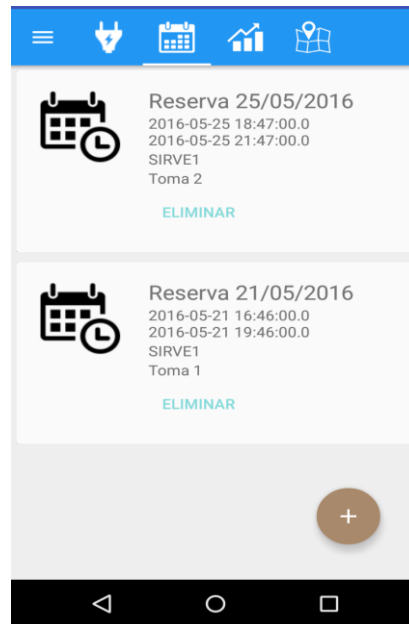


Figura 7. Aplicación: reservas.

2.5. Consulta de estadísticas

Esta pantalla resume el consumo eléctrico del cliente de dos formas. Como podemos ver en la figura 9, se muestra un gráfico de barras con los datos de consumo de energía (kWh), agrupados por meses. Por otro lado, se pueden visualizar las últimas operaciones realizadas y ver su información relevante (fecha, duración, nombre y tipo de cargador, y energía total recargada).

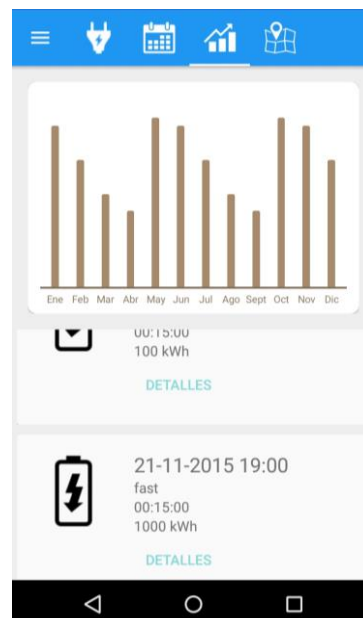


Figura 8. Aplicación: estadísticas y cargas realizadas.

2.6. Mapa de cargadores

Con el fin de que los usuarios encuentren con mayor facilidad y en cualquier momento los puntos de recarga que tiene disponibles, se incorpora en la aplicación un mapa proporcionado por la API de Google Maps, que localiza los cargadores disponibles que están autorizados para el usuario, así como la información necesaria para conocer su ubicación exacta y navegar hasta ella (localidad, dirección y coordenadas). Además, se puede seleccionar uno de los cargadores y activar la navegación hasta él a través de Google Maps.

Uno de los aspectos favorables de la aplicación web es la posibilidad de filtrar los cargadores en base a un radio de distancia en kilómetros respecto de la ubicación actual. En la parte inferior de la figura 10 se puede ver el slider que nos permite seleccionar el rango de alcance en el que se desea buscar. Así, los usuarios de la aplicación pueden ver fácilmente qué cargadores tienen cerca de su posición y elegir cual es el más conveniente en cuanto a facilidad de acceso, distancia o preferencias personales. En el caso de que los servicios de localización estén desactivados, se informa al usuario de que tiene que activarlos para conocer su ubicación actual y habilitar esta funcionalidad.

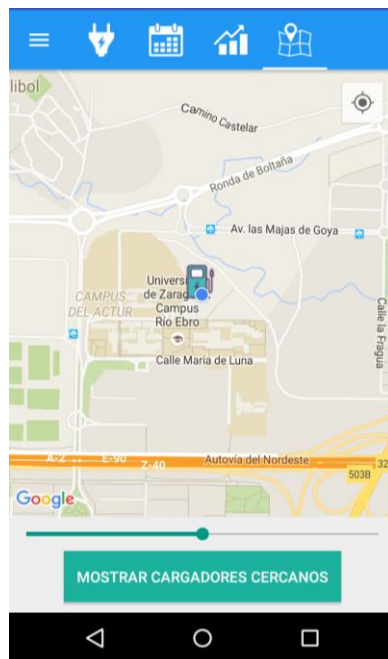


Figura 9. Aplicación: mapa de cargadores

3.Persistencia

Con el objetivo de que los usuarios no tengan que autenticarse cada vez que abren la aplicación, se ha implementado un sistema de persistencia que permite guardar los datos de la sesión en el dispositivo móvil con ayuda de la clase *Shared Preferences* [11].

Cada vez que se inicia la aplicación, se comprueba que estén cargados los datos del usuario en las variables globales. En caso contrario, se obtiene la información de las *Shared Preferences* o, en caso de no existir esa información, significará que el cliente había cerrado sesión o ésta había expirado por lo que es redireccionado directamente a la página de Login. El ciclo de vida de la aplicación y las comprobaciones realizadas en cada caso para guardar o recuperar las variables del usuario vienen explicadas en la tabla del anexo D.

Por otro lado, es importante que el proceso de recarga nunca se pierda y se retome en el caso de que la aplicación se cierre y vuelva a abrirse. Para ello se ha recurrido a un sistema similar al utilizado con los datos del usuario y se ha introducido una variable en el fichero de persistencia que indica si hay una recarga activa. El sistema comprueba esta variable cada vez que se inicia, recuperándola del fichero de persistencia en caso necesario, y dirige a la pantalla de recarga en curso, retomando las peticiones periódicas que informan de la energía suministrada al vehículo.

4.API móvil de los servicios web

Para el desarrollo de la API encargada de consumir los servicios web del Centro de Control se ha optado por utilizar la librería kSoap, que está formada por un parser y serializador de XML y la capa de transporte [12].

Los Servicios son llamados en segundo plano desde las actividades de la aplicación para no interferir el hilo principal. Se ha implementado una tarea asíncrona, llamada *AsyncTask* [13] en el Sistema Operativo Android, que recibe el nombre del servicio web a invocar y los parámetros necesarios en formato clave-valor. Esta clase construye el cuerpo del mensaje SOAP en base a estos parámetros y añade la cabecera de seguridad que se explicará en el siguiente capítulo. Además establece un timeout de quince segundos como máximo para recibir una respuesta del servidor. En el caso de que venza este tiempo o se produzca un fallo en la comunicación, la respuesta informará del error producido.

Esta tarea asíncrona no es llamada directamente desde la Activity sino que hay un *listener* que hace de intermediario y que se encarga de capturar la respuesta del *AsyncTask*, parsear el mensaje SOAP y devolver el objeto o respuesta final a la actividad principal, que actualiza la interfaz de usuario y notifica el resultado del servicio web.

Capítulo 6

Implementación de la seguridad de los servicios web

En este capítulo se describe la seguridad implementada para asegurar la comunicación de los servicios web entre la aplicación móvil y el servidor web.

1.Username Token

WS-Security permite el uso de distintos tokens de seguridad que extienden la funcionalidad y protección a nivel de mensaje y que son añadidos en la cabecera de seguridad. Para este trabajo se ha optado por usar un Username Token que permite de manera sencilla y sin tener que introducir una gran cantidad de modificaciones en las cabeceras, el transporte de los credenciales del usuario en el encabezado de seguridad del mensaje.

En un principio se pensó en utilizar el formato *#PasswordDigest* según el estándar OASIS [14]. Tal como se puede observar en la figura 10, este método incluye en el token el *timestamp*, un *nonce* o número aleatorio y un *digest* computado con la contraseña y los dos elementos anteriores, además del nombre de usuario. Esto evita tener que gestionar la contraseña como texto y frena ataques de respuesta manteniendo en caché los *nonces* y asignándoles un tiempo de frescura en base al timestamp. Sin embargo, este método no era factible debido a que en el servidor no se almacena la contraseña en texto plano y por lo tanto no era posible comparar ambos códigos Hash a no ser que se cambiase la base de datos del servidor para que almacenase también este valor.

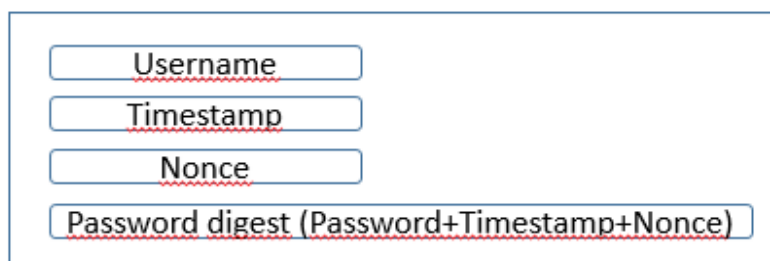


Figura 10. Username token con formato PasswordDigest.

Finalmente, se decidió implementar el Username Token con la contraseña en formato *#PasswordText* según el estándar [14], es decir, como texto plano. De acuerdo con la especificación WS-SecurityPolicy [15], se ha especificado en la política de los servicios web, el fichero XML que define la estructura obligatoria de los mensajes SOAP de entrada y salida, la aparición obligatoria de este Token en todo mensaje que vaya desde el iniciador de la petición (la aplicación cliente) hasta el receptor de ésta (el servidor o Centro de Control).

Además, también es necesario añadir un elemento *<ramp:RampartConfig>* a la política. Este elemento proporciona extensiones específicas para la gestión de Rampart y se utiliza para definir la clase encargada de realizar las llamadas de retorno del token de usuario. Esta clase extrae la información relativa al usuario y contraseña del *Username Token* y verifica que los datos coinciden con los que se encuentran almacenados en la base de datos.

De esta forma, dado que todo mensaje (figura 11) irá acompañado de este Token, se asegura qué usuario es el que está solicitando el servicio y se restringe la llegada de mensajes que no vayan acompañados con esta información y, por lo tanto, posibles peticiones enviadas desde dispositivos ajenos al sistema no registrados como clientes.

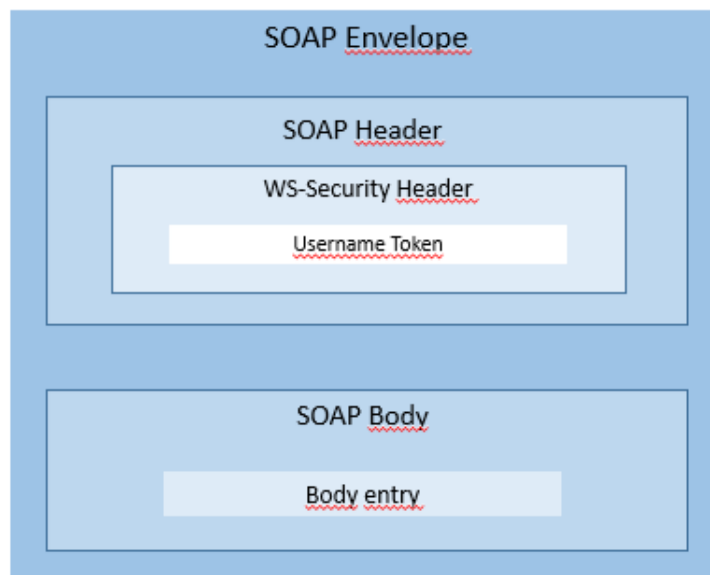


Figura 11. Mensaje SOAP.

2. Asegurando la comunicación

Sin ninguna otra medida de seguridad, cualquier persona capaz de capturar el mensaje enviado entre el cliente y servidor puede ver al completo el contenido del mensaje, incluyendo el Username Token, que contiene las credenciales del usuario. Por esta razón, se propusieron distintas soluciones para incrementar la seguridad y que se detallan a continuación.

2.1. Alternativa inicial: seguridad a nivel de mensaje

En un primer momento se pensó en implementar seguridad a nivel de mensaje, contemplada gracias al módulo de seguridad WS-Security, que permite introducir en los mensajes SOAP claves encriptadas, firmas y certificados, entre otros.

Este método implementaba un algoritmo de doble cifrado en el que, a priori, se daba por hecho que el cliente tenía la clave pública del servidor web y que está ilustrado en la figura 12. El cliente, es decir, la aplicación móvil, cifraba con su clave privada el cuerpo del mensaje para generar la firma digital y, por otro lado, lo encriptaba con una clave secreta generada. Además del cuerpo del mensaje, encriptado y firmado, el cliente enviaba en el mensaje SOAP la clave secreta encriptada con la clave pública del servidor y su certificado público, para poder verificar la firma digital. Cuando el Centro de Control recibía el mensaje, desencriptaba la clave secreta con su clave privada y la utilizaba para leer el cuerpo del mensaje. Después, comprobaba la firma del remitente con el certificado público del cliente incluido en el mensaje [16]. La construcción de la política y un ejemplo de mensaje con este algoritmo puede verse en la sección 1 del anexo C.

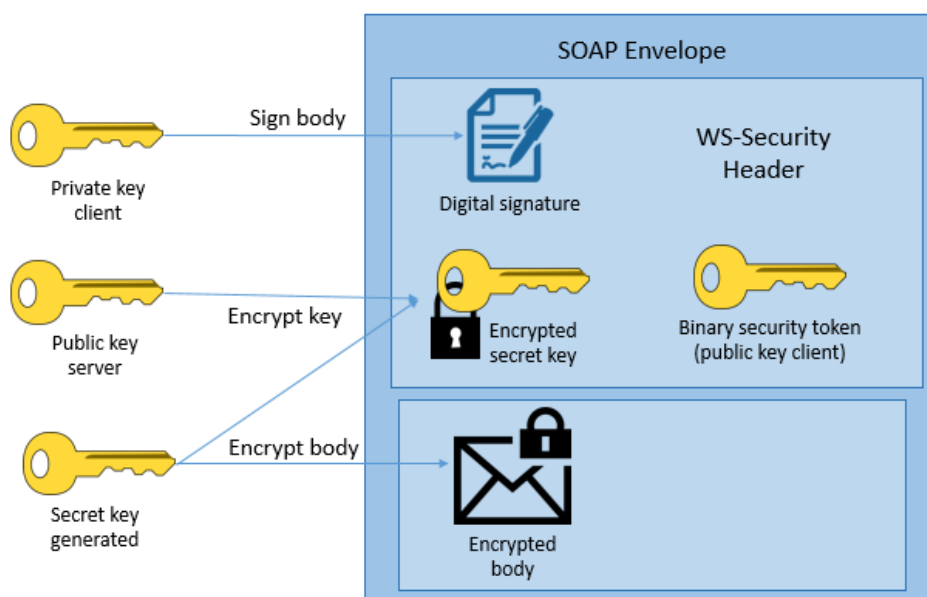


Figura 12. Algoritmo de doble cifrado con WS-Security.

Esta alternativa se desechó ya que no se encontró documentación relativa a la implementación de WS-Security en Android y la solución pasaba por construir los mensajes y cifrar las etiquetas manualmente. Esto obligaba a añadir librerías externas a Android que, en ocasiones, no eran compatibles y aumentaban en gran medida el peso de la aplicación. De esta manera, dado que no era el objetivo del proyecto, se decidió replantear la solución propuesta para implementar la seguridad y enfocarla a la capa de transporte, como veremos en la siguiente sección.

2.2. Implementación final: cifrado de la contraseña y seguridad a nivel de capa de transporte

La decisión final a la hora de implementar la seguridad de los servicios fue utilizar un canal de comunicación cifrado, utilizando el protocolo HTTPS sobre SSL e impidiendo que cualquiera que intente monitorizar el mensaje pueda ver su contenido. Esta implementación se sumaba a la gestión de los credenciales a nivel de mensaje, que se consigue a través del *Username Token* explicado en la primera sección de este capítulo. Según la documentación de *WS-SecurityPolicy* [15], es necesario añadir un elemento `<sp:TransportBinding>` que contiene la información relativa a la capa de transporte. Dentro de esta etiqueta hace falta especificar el tipo de token utilizado, `Https` en nuestro caso. La especificación de la política así como un ejemplo de un mensaje que la cumpla pueden visualizarse en la sección 2 del anexo C.

Para utilizar este canal cifrado es necesario configurar el servidor web para que soporte este protocolo. Dado que se usa Apache Tomcat como servidor, es necesario modificar el fichero de configuración `server.xml` para abrir un conector que acepte conexiones `https` y especificar el `keystore` correspondiente, tal como se muestra en la figura 13.

```
<ConnectorSSLEnabled="true" clientAuth="false"  
  keystoreFile="C:/XX/XX/keystore" maxThreads="150" port="8443"  
  protocol="HTTP/1.1" scheme="https" secure="true" sslProtocol="TLS"/>
```

Figura 13. Configuración SSL de Tomcat.

Por otro lado, es necesario almacenar la contraseña del usuario en el dispositivo móvil para poder incluirla en la cabecera de las peticiones que realice, sin tener que recurrir a escribirla cada vez que se quiera realizar una operación. Guardar la contraseña en texto plano no es una buena práctica ya que sería totalmente vulnerable en el caso de que el móvil del usuario fuera robado y la persona que lo poseyera tuviese los conocimientos necesarios para hacer root y acceder a los archivos ocultos [17].

Para solucionar este problema, se ha decidido guardar en el dispositivo móvil la contraseña cifrada con una clave conocida tanto por la aplicación como por el servidor web. De esta manera, se almacena la contraseña en el móvil tras ser encriptada con la clave pública y será utilizada en cada petición como puede observarse en la figura 14. El módulo Rampart del servidor web, por su parte, recibirá los mensajes y utilizará la clave privada para descifrar la contraseña contenida en el Username Token y poder autenticar al usuario.

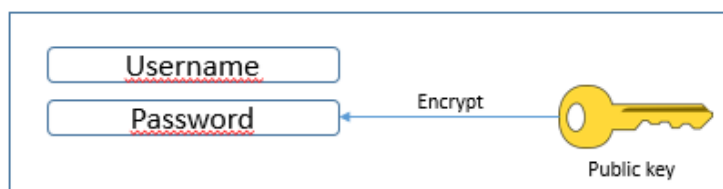


Figura 14. Username Token con formato PasswordText.

3. La configuración de la seguridad en Android

Como se ha comentado en el apartado 4 del capítulo 5, se ha utilizado la librería KSoap para la integración del cliente de servicios web en la aplicación. A nivel de seguridad esta librería no proporciona herramientas abiertamente a excepción de la posibilidad de utilizar HTTPS como protocolo de comunicación.

A la hora de crear las cabeceras de seguridad es necesario construir los mensajes paso a paso, especificando los elementos de cada nivel y los prefijos y URIs correspondientes a cada espacio de nombres. Para este cometido se ha creado una clase que permite añadir al envelope cualquier tipo de cabecera de seguridad como Tokens, claves encriptadas o referencias, entre otros.

En cuanto a la comunicación, se ha utilizado la clase *HttpstransportSE*, que necesita como parámetros la dirección del servicio, el puerto, la dirección donde está contenido el WSDL y el *timeout* de la conexión. Por último, uno de los problemas que surgen al utilizar seguridad en la capa de transporte a través de SSL reside en la necesidad de confiar en los certificados. Dado que el certificado del servidor web no está totalmente en producción y es autofirmado, Android no lo admite en su nivel de confianza y lo rechaza, lanzando una excepción cuando recibe la respuesta. Para solucionarlo se ha optado por establecer un gestor de confianza que es activado antes de iniciar una comunicación SSL y permite confiar en certificados no oficiales.

Capítulo 7

Pruebas y tests

En este apartado se describen las distintas pruebas realizadas para verificar el código, además de aquellas realizadas a usuarios para evaluar la aplicación.

1. Pruebas unitarias

Conforme el proyecto iba creciendo, se hizo patente la necesidad de crear y automatizar una serie de test que validaran el correcto funcionamiento ante cualquier cambio introducido en el código y así poder detectar posibles errores rápidamente.

Los test unitarios corren pequeños módulos aislados, que pueden ser desde una clase a un método y pueden ser ejecutados en la máquina local, sin necesidad de un emulador o dispositivo, ya que no interactúan con otras partes del sistema [18]. Esto favorece el desarrollo ágil y la validación inmediata por separado de cualquier pieza de código ante una modificación. En el caso de métodos que interactúen con servicios web, se ha realizado un *mock*, que aísla la unidad de otras dependencias a través de la librería Mockito, y permite asignar a un servicio la devolución deseada. De este modo, se puede testear el comportamiento de la aplicación ante cualquier tipo de situación y/o respuesta. La relación completa de pruebas unitarias realizadas para testear este trabajo puede verse recogida en la sección 1 del anexo E.

2. Pruebas de integración

Las pruebas de integración para valorar el correcto funcionamiento de la aplicación en conjunto con otros sistemas (base de datos y servicios web) no se han automatizado y se han testeado manualmente siguiendo una lista de casos de uso con la ayuda de un teléfono móvil con conexión a internet y que se puede ver en detalle en la sección 2 del anexo E.

3. Test de usabilidad

Una vez se alcanzó la primera versión estable de la aplicación, se decidió realizar una serie de pruebas de usabilidad para evaluar el trabajo a través de usuarios potenciales. Para ello, se seleccionó un grupo de cuatro personas de distinta edad y ámbito de trabajo pero con un elemento en común que hacía significativo el estudio: estar familiarizados con el vehículo eléctrico y el sistema de recarga. Este test proporcionaba una visión imposible de obtener por los desarrolladores o supervisores que han estado en contacto durante todo el proceso de diseño e implementación y una información directa de cómo utilizan el sistema usuarios reales, además de qué dificultades o problemas pueden tener cuando navegan por él. Esta información es utilizada para evaluar la capacidad de la aplicación para satisfacer y cumplir los requisitos establecidos al inicio del proyecto.

La prueba fue realizada con un móvil Motorola Moto G bajo supervisión directa y consistió en una serie de tareas que los usuarios tenían que llevar a cabo a través de la aplicación y que cubrían en su totalidad las funcionalidades y requisitos del proyecto. De esta manera, se evaluó el tiempo de respuesta, la capacidad de reacción, y la satisfacción y concordancia con el resultado esperado de cada funcionalidad de la aplicación. Además, se utilizaba un modelo de pensamiento retrospectivo en voz alta en el que el usuario, al acabar la tarea, comentaba cómo se había encontrado con el sistema, qué problemas había tenido y qué mejoras consideraría oportunas [19]. En la sección 3 del anexo E se puede ver más detalladas todas las acciones que fueron solicitadas a los usuarios y que permitieron valorar la usabilidad del sistema.

3.1. Resultados

Todos los usuarios han coincidido en que la aplicación les sería de gran utilidad en el caso de poseer un vehículo eléctrico y que la usarían en su día a día para agilizar y realizar las recargas de vehículo con mayor comodidad. El hecho de poder acceder a esta información en cualquier momento y lugar a través del dispositivo móvil lo consideran esencial hoy en día y ayudaría a no tener que esperar junto al vehículo al realizar cargas de larga duración. Pese a ello, también han quedado patentes varias situaciones en las que los usuarios han tenido más dificultades y han propuesto una presentación distinta de los contenidos para facilitar su comprensión y uso o proporcionar una información de mayor utilidad. De esta manera, se pueden revisar los diferentes problemas para mejorar la usabilidad e introducir los cambios necesarios en las próximas versiones de la aplicación (figura 15) [19].

Pantalla	Problema / Posible mejora	Severidad	Solución
Mapa de cargadores	No se ve clara la selección del rango	Alta	Poner un rótulo explicativo o una leyenda que facilite la comprensión
Mapa de cargadores	Información más detallada de las estaciones	Leve	Incluir las tomas con las que cuenta y dar la posibilidad de reservar directamente
Mapa de cargadores	Al cambiar el rango de kilómetros no se aprecia nada en el mapa	Leve	Cambiar dinámicamente el zoom del mapa en base al rango de kilómetros seleccionado
Gestión de recargas	Captura del código QR ininterrumpida	Moderada	Poner un tiempo de espera, mostrando el error, hasta volver a escanear el código
Estadísticas	Filtrar por periodo de tiempo	Leve	Permitir hacer click sobre el mes deseado para ampliar la información de consumo

Figura 15. Tabla-resumen del test de usabilidad.

Capítulo 8

Conclusiones y líneas futuras

1. Resultados finales

El trabajo ha alcanzado el objetivo y se ha conseguido una versión estable de la aplicación cumpliendo todos los requisitos establecidos. En un primer momento, se estableció como requisito único y principal de la aplicación que permitiese iniciar recargas de vehículo eléctrico remotamente desde el móvil a través de servicios web y tuviese la posibilidad de monitorizar la carga para poder ver el progreso actual. Sin embargo, poco a poco estos requisitos se fueron ampliando para abarcar otras funcionalidades que se consideraron interesantes para hacer la aplicación más atractiva y llegar a un público más numeroso.

De esta manera, la aplicación cuenta con un sistema de gestión de recargas que funciona mediante el escaneo de un código QR, tecnología que atrae a posibles usuarios debido a su facilidad de uso y al hecho de evitar que sea necesario disponer de una tarjeta RFID para autenticarse en el punto de recarga. Por otro lado, se han desarrollado funcionalidades extra que amplían la experiencia de uso del sistema de recarga como son la gestión de reservas, consulta de estadísticas o búsqueda de estaciones de recarga en el mapa.

Otro de los objetivos que se propusieron una vez comenzado el proyecto y que se ha completado con éxito ha sido diseñar e implementar una seguridad mínima en los servicios web que facilitase enviar y recibir mensajes SOAP dentro de una canal de comunicación seguro y cifrado, y permitiese validar los mensajes recibidos por el servidor web. Además de evitar que se pudiesen monitorear los mensajes y ver su contenido, este sistema tenía la obligación de asegurar la identidad de los remitentes de los mensajes y así poder asignar correctamente las operaciones de recarga o reservas de estaciones eléctricas a los usuarios correspondientes, evitando posibles fraudes y ataques ajenos que pusieran en riesgo el sistema y los clientes.

Para concluir, la aplicación móvil, aunque no se prevé que sea a corto plazo, está lista para ponerse en producción a falta de ligeras modificaciones que permitan adecuarse al sistema en el que se integre o al modelo de negocio y la seguridad necesaria para garantizar su confidencialidad.

2. Conclusión personal

Antes de empezar mi Trabajo de Fin de Grado tenía claro que quería realizarlo en el ámbito empresarial. De esta manera, tenía la oportunidad de introducirme en el mundo laboral a la vez que aumentaba mis conocimientos y realizaba un trabajo que plasmara lo aprendido durante los últimos cuatro años. Elegí la fundación CIRCE porque me atrajo este proyecto, su contexto temático y la libertad para introducir características o mejoras en base a las herramientas y datos de los que dispusiese. Además, se trataba de un proyecto real, con proyección de ponerse en producción y que interactuaba con otros sistemas ya desarrollados con los que había que lidiar para el buen funcionamiento del sistema global, un reto que no había afrontado con anterioridad.

En cuanto a la tecnología y herramientas que iba a necesitar, lo único en lo que tenía dominio suficiente para empezar a trabajar era el desarrollo móvil ya que había realizado pequeños proyectos con anterioridad. Por eso, el reto de profundizar en mis conocimientos en Android y, además, ampliar mis habilidades con el desarrollo de servicios web tanto en su implementación como en asegurar las comunicaciones, hizo que me decantara por realizar este trabajo. A pesar de que la temática no está directamente relacionada con la rama que he cursado (Computación) esta experiencia me ha servido para conocer más de cerca uno de los campos de trabajo más amplio como es el desarrollo de Software y todas sus etapas. Además, he disfrutado de un buen entorno de trabajo que ha facilitado mi progreso y en el que he aprendido buenas prácticas y métodos de trabajo de otros profesionales.

Durante todo el desarrollo he sido muy crítico con mi trabajo y he intentado limar hasta el más mínimo detalle para explotar al máximo mis cualidades y entregar el mejor trabajo posible. Esto me ha llevado a dedicar más tiempo del que debería a partes del trabajo que quería incluir y que quizás debería simplificado cumpliendo de igual manera los requisitos. Es el caso de la implementación de seguridad en los servicios web, uno de los temas más novedosos para mí y que me ha llevado, en proporción, una gran cantidad de tiempo en el intento de implementar seguridad a nivel de mensaje en Android. A pesar de ello, el conocimiento que he adquirido en un tema que se ha descubierto para mí es muy interesante para desarrollarme en un futuro. Todas las lecciones, errores y aciertos sirven para aprender, progresar y poder manejar estas situaciones en proyectos futuros.

3. Líneas futuras

Como se ha comentado anteriormente en los resultados, no está previsto que la aplicación sea introducida en producción a corto plazo pero sí que hay posibilidades de que esté en proyectos futuros o que sea comercializada a medio-largo plazo. A continuación se detallan las posibles líneas a nivel de proyectos en las que podría tener cabida este trabajo:

- El potencial de la aplicación es muy grande ya que permitirá controlar remotamente las cargas de vehículo eléctrico, así como toda la información en relación a las infraestructuras y estadísticas de consumo, y es por ello que varias empresas del sector que cuentan con un gran número de puntos de recarga eléctrica se han interesado y meditan incorporarlo a sus sistemas junto al Centro de Control desarrollado también en la fundación.
- Asimismo, se está colaborando en el proyecto Flexiciency [20], promovido por el H2020, programa más grande de investigación e innovación de Europa, y que está relacionado con Smart Cities y la gestión de la calidad y cantidad de información proporcionada acerca de los consumos energéticos. En este proyecto será necesario integrar todo el control del consumo y automatización de edificios, cargadores o paneles solares en la aplicación, pudiendo convivir con el sistema actual o en paralelo a éste, y poniendo al alcance del usuario tener en sus manos toda la información acerca de su hogar y su vehículo eléctrico.

Por otro lado, quedan pendientes diferentes mejoras en la aplicación que se han propuesto tras las pruebas de usabilidad o que serían necesarias en el caso de cambiar el modelo de negocio o gestión:

- Mejores a nivel de interfaz surgidas a raíz de las propuestas de los usuarios en las pruebas de usabilidad:
 - Implementar de manera más intuitiva la selección del rango de kilómetros, además de cambiar el mapa dinámicamente.
 - Aportar más información en el mapa de cargadores acerca de las tomas, así como permitir la reserva directa de cargadores.
 - Permitir filtrar las estadísticas por periodos de tiempo.
 - Visualizar los huecos libres de las estaciones de recarga.

- Se ha dejado abierta la oportunidad de incrementar la seguridad a nivel de mensaje con el módulo de seguridad utilizado en el desarrollo de los servicios web, WS-Security. Está previsto que en un futuro se integren pagos electrónicos en la aplicación para que los usuarios puedan recargar su saldo y utilizar las estaciones de carga sin tener que recurrir a otros medios. Para este caso de uso sería interesante implementar un algoritmo de doble cifrado, expuesto en el capítulo 6 y ejemplificado en el anexo C, sección 1.
- Utilizar certificados digitales validados por una entidad certificadora. De esta forma se garantiza ante terceros que la información que se transmite en la operación electrónica es totalmente confidencial entre los usuarios y las entidades que la llevan a cabo mediante la vinculación entre la identidad (el servidor web) y su clave pública [21].

Capítulo 9

Bibliografía

- [1] Urbener. Proyecto SIRVE. Disponible en: <https://www.urbener.com/sirve/>.
- [2] Endesa. “Recarga del vehículo eléctrico”. Disponible en:
<http://www.endesavehiculoelectrico.com/vehiculo-electrico/recarga>
- [3] International Data Corporation “Smartphone OS Market Share, 2015 Q2”
Disponible en: <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>.
- [4] Android Developers. “Material Design”. Disponible en:
<https://developer.android.com/design/material/index.html>.
- [5] Open Charge Alliance. “OCPP v1.5 A functional description”. 2012
- [6] Microsoft Developer Network. “Message Security”. Disponible en:
<https://msdn.microsoft.com/en-us/library/ms733137.aspx>
- [7] Apache Software Foundation. “Apache Axis2/Java”. Disponible en:
<http://axis.apache.org/axis2/java/core/>
- [8] Lahiru Sandakith. “Creating Bottom Up Web Service via Apache Axis2”. 2007.
Disponible en:
http://www.eclipse.org/webtools/community/tutorials/BottomUpAxis2WebService/bu_tutorial.html.
- [9] Android Developers. “Using ViewPager for Screen Slides”. Disponible en:
<https://developer.android.com/training/animation/screen-slide.html>
- [10] Sean Farrel. “Choosing the Right Background Scheduler in Android”. 2016.
Disponible en: <https://www.bignerdranch.com/blog/choosing-the-right-background-scheduler-in-android/>.
- [11] Android Developers “Shared Preferences”. Disponible en:
<https://developer.android.com/reference/android/content/SharedPreferences.html>.

- [12] kSOAP2 Project. "Lightweight, efficient SOAP on Android". Disponible en: <http://simpligility.github.io/ksoap2-android/index.html>.
- [13] Android Developers "AsyncTask". Disponible en: <https://developer.android.com/reference/android/os/AsyncTask.html>.
- [14] OASIS Standard. "Web Services Security Username Token Profile Version 1.1.1". 2012.
- [15] OASIS Standard. "WS-SecurityPolicy 1.2". 2007.
- [16] Dennis Sosnoski. "Java Web services: Axis2 WS-Security signing and encryption". IBM DeveloperWorks. 2009. Disponible en: <http://www.ibm.com/developerworks/library/j-jws5/>
- [17] Rémi Pradal. "Develop a secured Android Application". 2015. Disponible en: <http://blog.octo.com/en/develop-secured-android-application/>.
- [18] Android. "Test your App. Testing concepts". Disponible en: <https://developer.android.com/studio/test/index.html>.
- [19] Usability.gob. "Usability Testing". Disponible en: <https://www.usability.gov/how-to-and-tools/methods/running-usability-tests.html>
- [20] H2020. "Flexiciency Project". Disponible en: <http://www.flexiciency-h2020.eu/project-overview/mission>.
- [21] Registro de Empresas Acreditadas. "Certificados digitales y Entidades Certificadoras". Disponible en: http://rea.mtin.gob.es/rea/pub/rea_ca.html.

Anexo A

Gestión del proyecto

Como se ha comentado en la memoria del trabajo, se ha seguido un desarrollo ágil debido a que los requisitos no eran inamovibles y se han ido refinando con el paso del tiempo hasta llegar al producto final. En la figura 1 se ha plasmado el tiempo de dedicación y la distribución aproximada de las distintas tareas a lo largo del tiempo a través de un diagrama de Gantt. A pesar de que se ilustre como un desarrollo en cascada, desde cualquiera de las fases se han realizado iteraciones hacia atrás ante cambios de requisitos o de diseño pero, al ser periodos más cortos, se han considerado como parte de las etapas generales. A continuación se explican brevemente las distintas etapas por las que se ha pasado en el desarrollo de este trabajo:

- Fase de análisis: En esta etapa se han especificado los primeros requisitos del proyecto, primeros bocetos y diagramas. Se ha vuelto a esta fase en varios puntos del desarrollo en los que ha sido necesario redefinir los análisis para introducir los cambios pertinentes. Horas aproximadas: 50h.
- Fases de aprendizaje: Antes de empezar una fase se analizaba qué herramientas había que utilizar y cuáles de ellas necesitaban un aprendizaje previo para poder comenzar. Horas aproximadas: 90h.
- Desarrollo: Como era de esperar, el desarrollo de la aplicación y servicios web ha sido la fase más larga de este Trabajo Fin de Grado. Horas aproximadas: 200h.
- Pruebas: Esta fase se ha encontrado prácticamente al final de cualquier momento del desarrollo aunque con más exhaustividad en la fase final del proyecto. Horas aproximadas: 55h.
- Implementación de seguridad: La implementación de la seguridad ha sufrido numerosos cambios que se han ido redefiniendo en las fases finales del desarrollo. Horas aproximadas: 150h.
- Redacción de la memoria. Horas aproximadas: 40h.

Total horas = 585

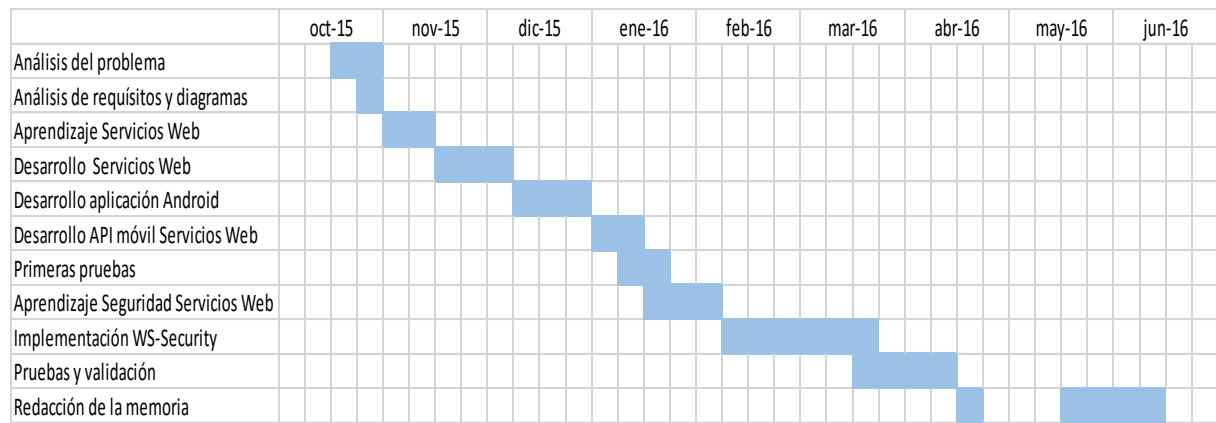


Figura 16. Diagrama de distribución de trabajo.

Anexo B

Análisis y diagramas del trabajo

En este anexo se presentan los diagramas realizados en la fase de análisis de este Trabajo Fin de Grado, además de los primeros bocetos.

1. Primer prototipo en papel

En la fase inicial del proyecto se realizó un primer boceto en papel de la aplicación y las pantallas correspondientes a la gestión de la recarga de vehículo eléctrico. A partir de esta versión se han ido desarrollando el resto de prototipos hasta dar con la aplicación final, que podemos ver más adelante en el mapa de navegación.

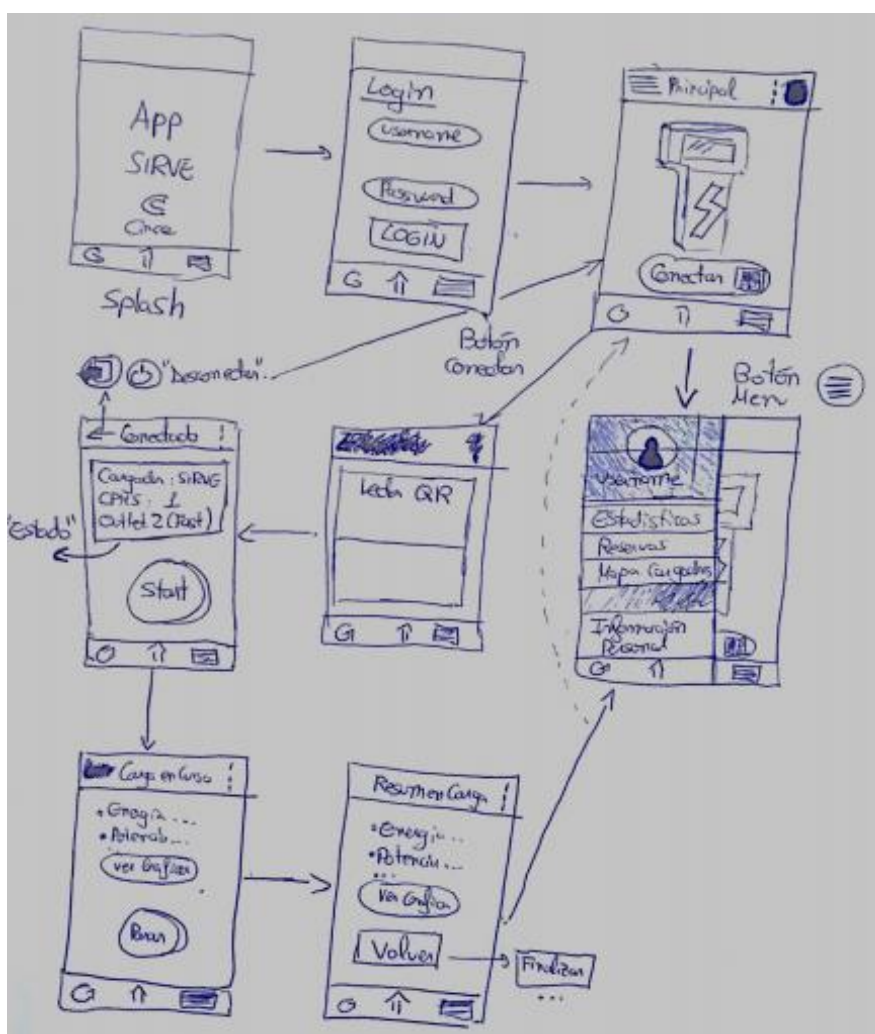


Figura 17. Primer prototipo en papel.

2. Diagrama de casos de uso

Este diagrama representa la forma en la que el actor, usuario de la aplicación en este caso, opera con el sistema, especificando las relaciones entre los casos de uso y el actor, además de las dependencias entre los casos de uso (*include* y *extends*).

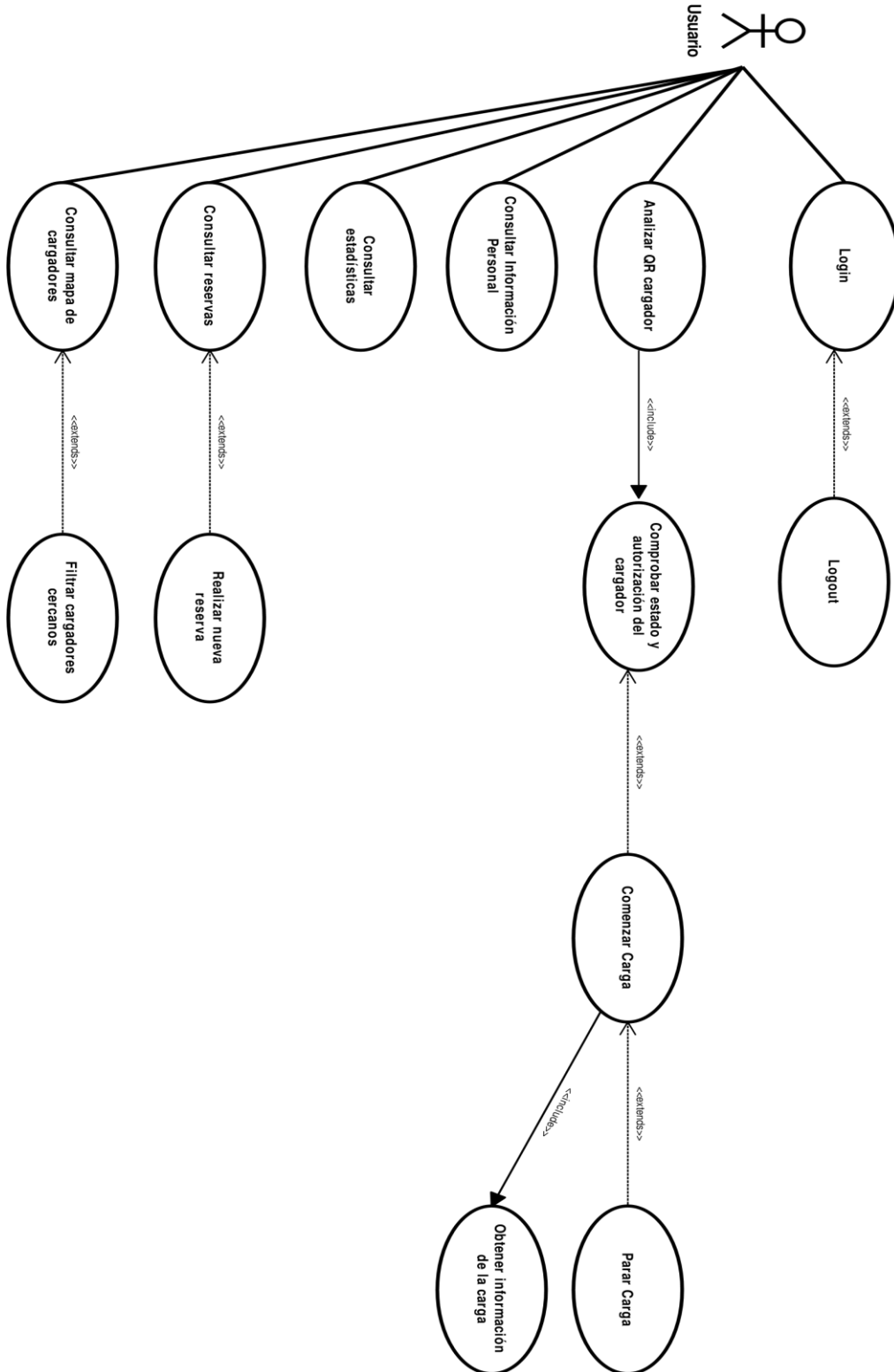


Figura 18. Diagrama de casos de uso.

3. Diagramas de secuencia

Estos diagramas muestran la interacción de un conjunto de objetos en la aplicación a través del tiempo y están modelados para tres casos de uso: realizar una nueva reserva, iniciar una recarga y autenticarse en la aplicación. Existen cinco objetos distintos que interactúan entre sí: los usuarios, que manejan la aplicación Android, y ésta a su vez se comunica a través de servicios web con el servidor. Finalmente el Centro de Control intercambia información con la base de datos y, en el caso de recargas de vehículo, también se comunica con las Smartbox de los cargadores.

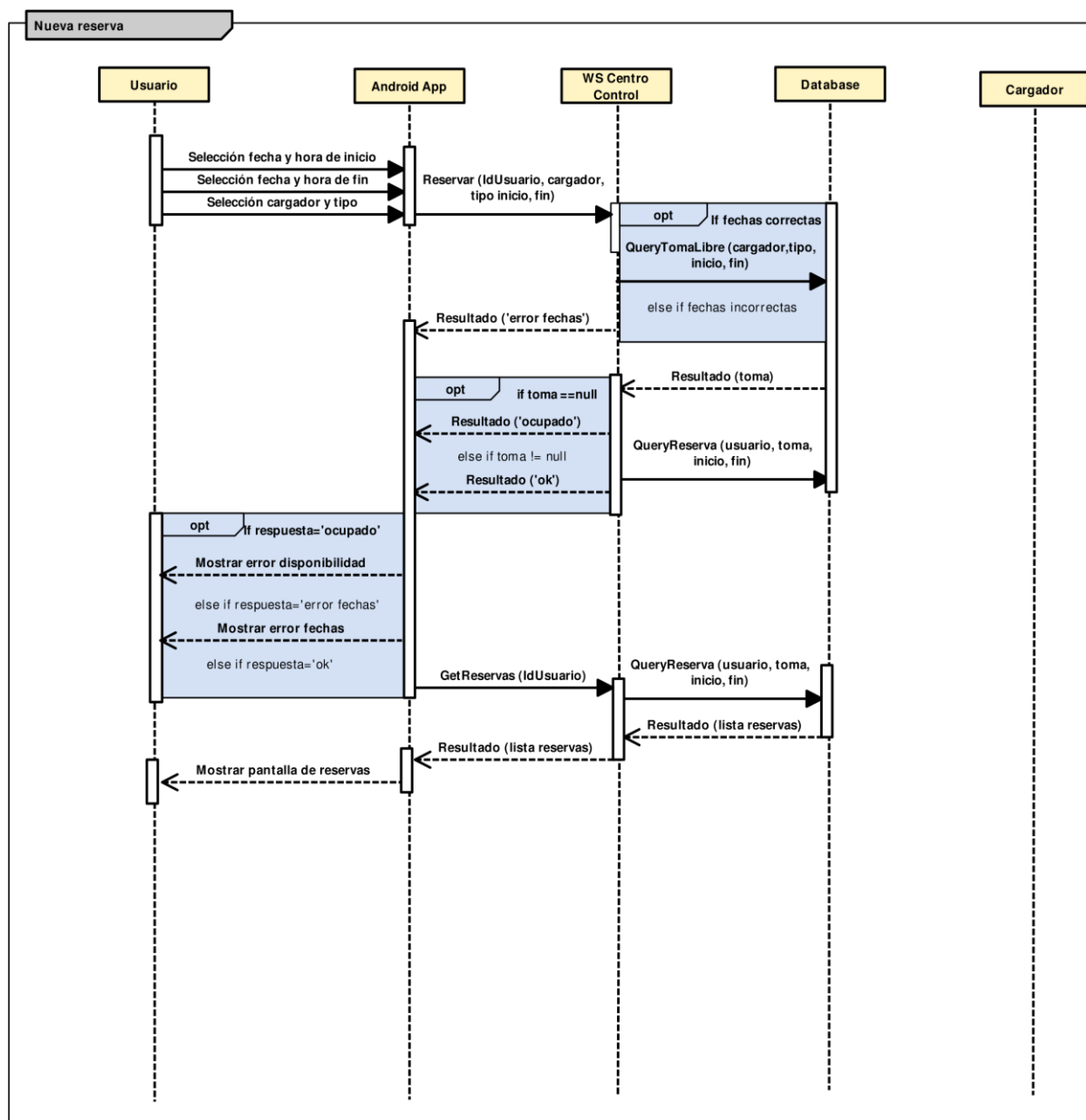


Figura 19. Diagrama de secuencia: nueva reserva.

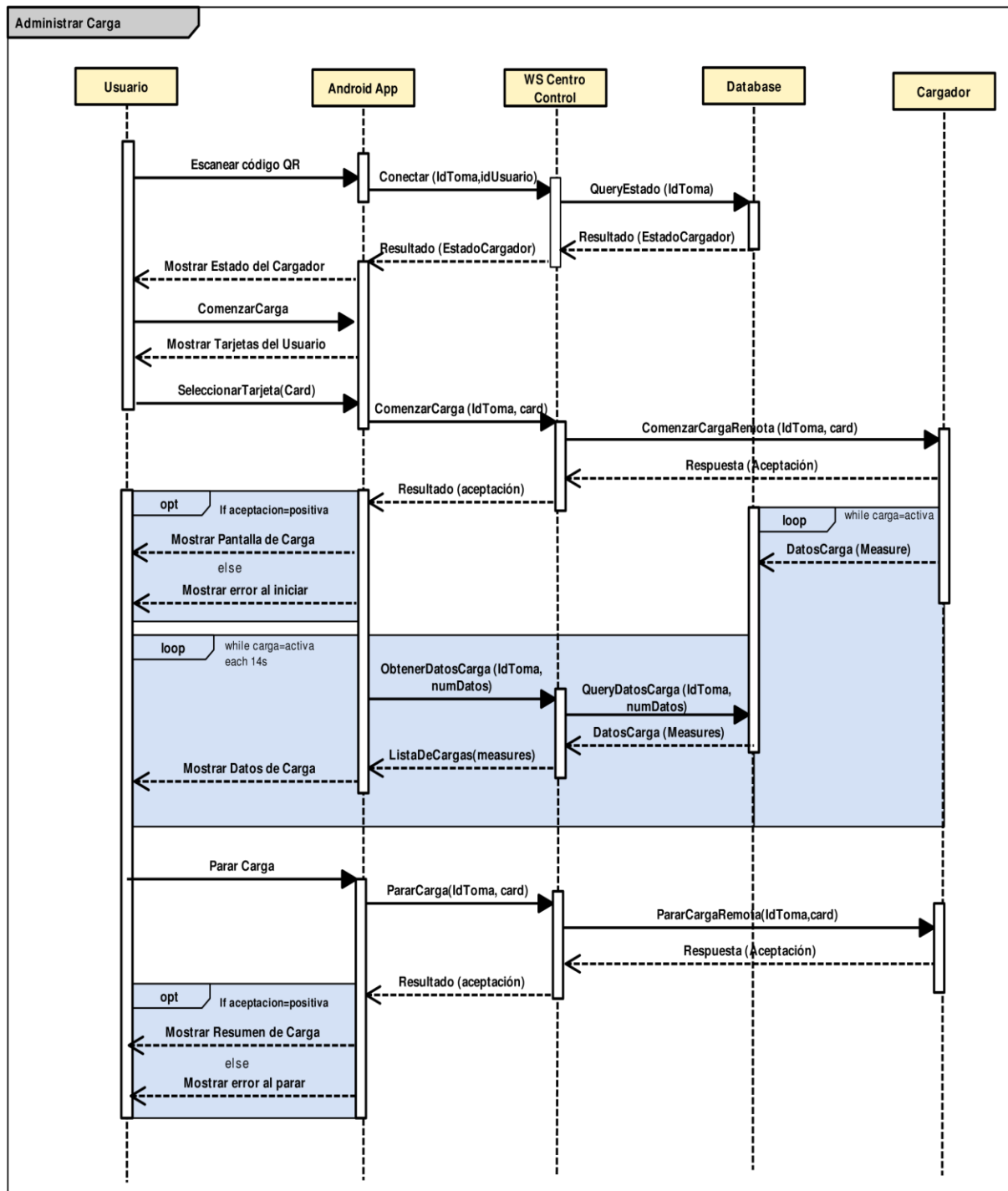


Figura 20. Diagrama de secuencia: monitorizar recargas.

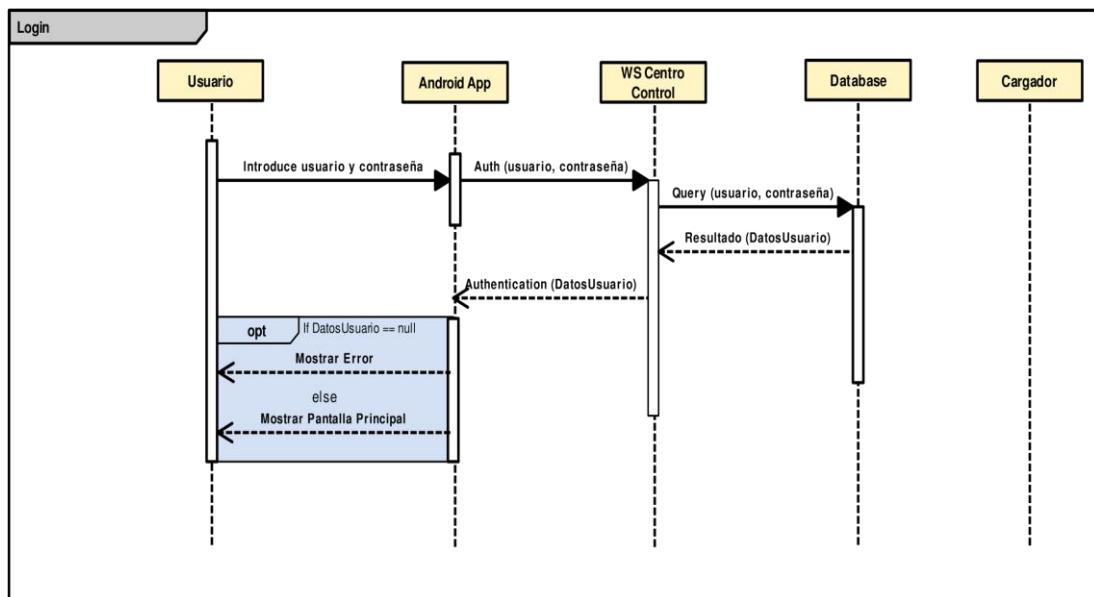


Figura 21. Diagrama de secuencia: login.

4. Mapas de navegación

El mapa de navegación permite ver la secuencia lógica de pantallas y acciones por las que tiene que pasar el usuario para interactuar con el sistema y realizar una tarea determinada. A continuación se presenta el recorrido necesario para llevar a cabo las principales operaciones de esta aplicación: iniciar y monitorizar recargas, realizar reservas, autenticarse en la aplicación y buscar cargadores cercanos en un radio determinado.

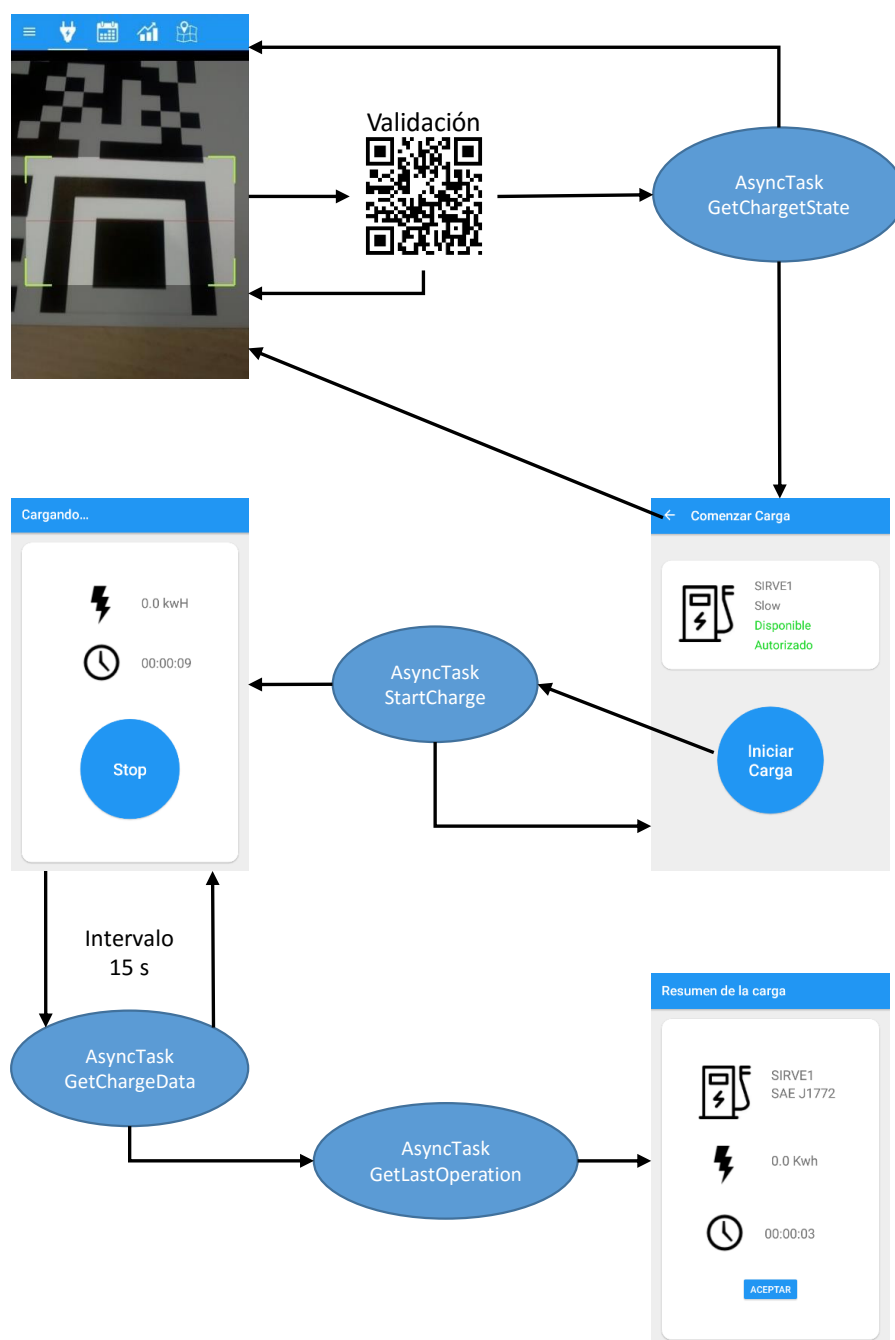


Figura 22. Mapa de navegación: monitorizar una recarga.

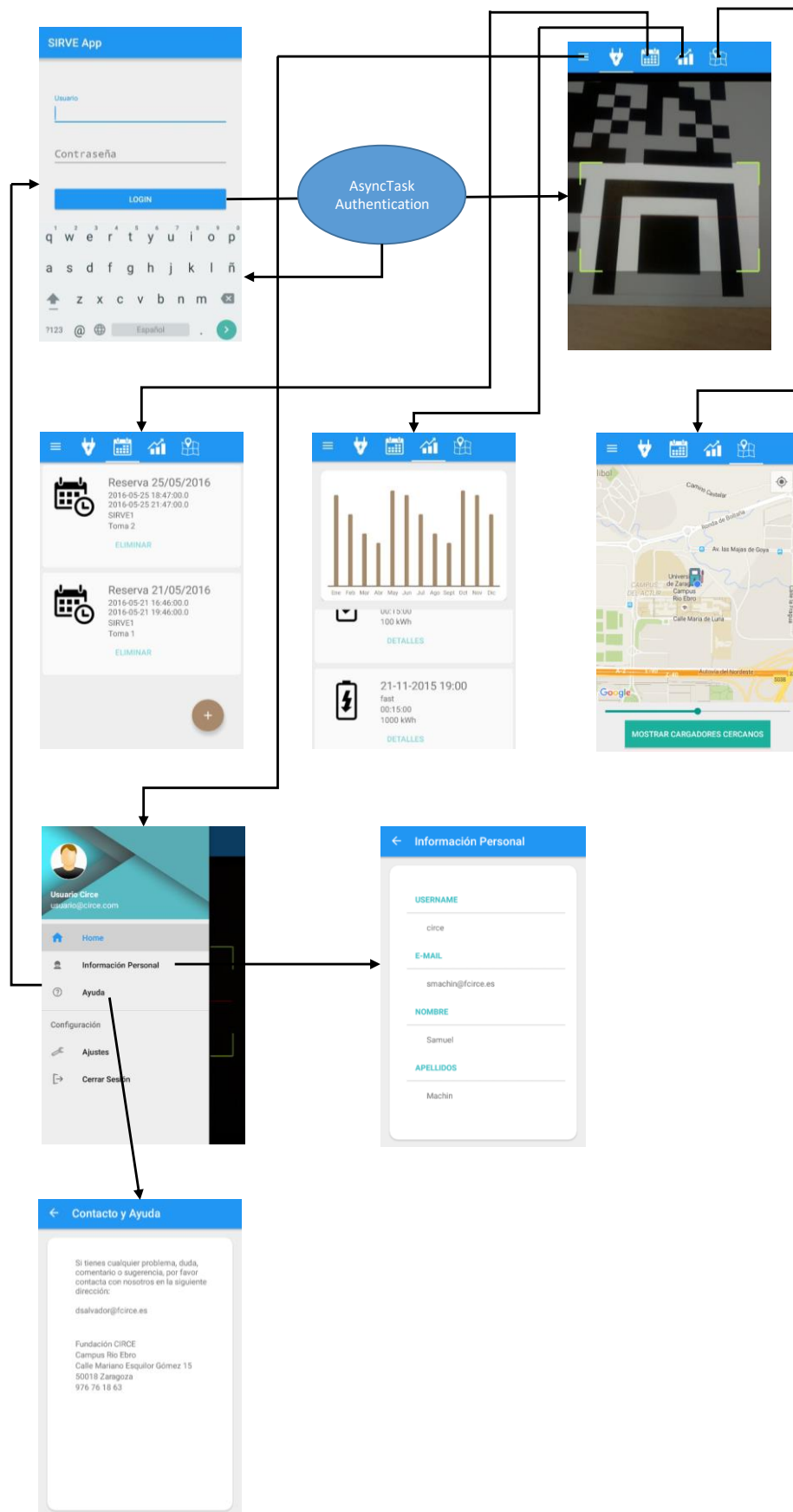


Figura 23. Mapa de navegación: login y menú principal.

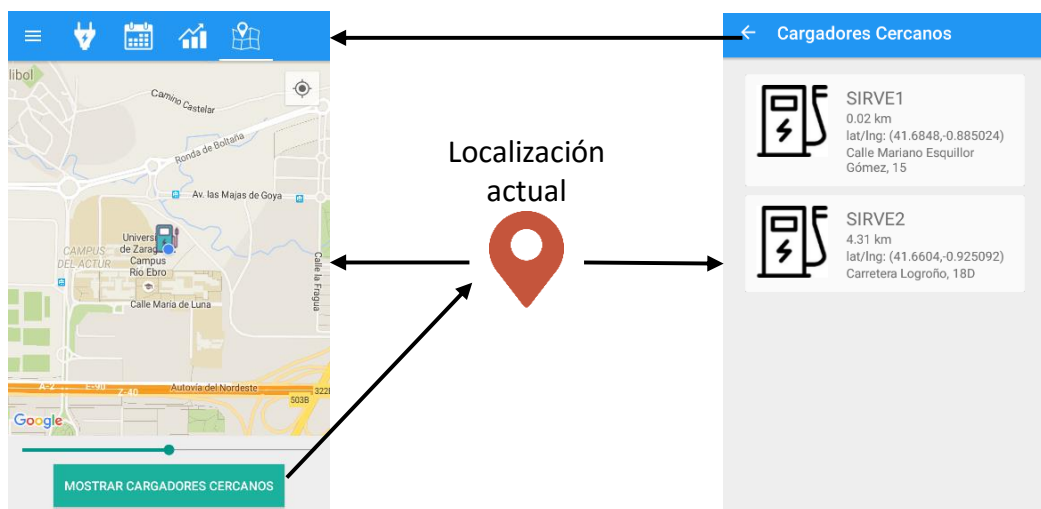


Figura 24. Mapa de navegación: mostrar cargadores cercanos.

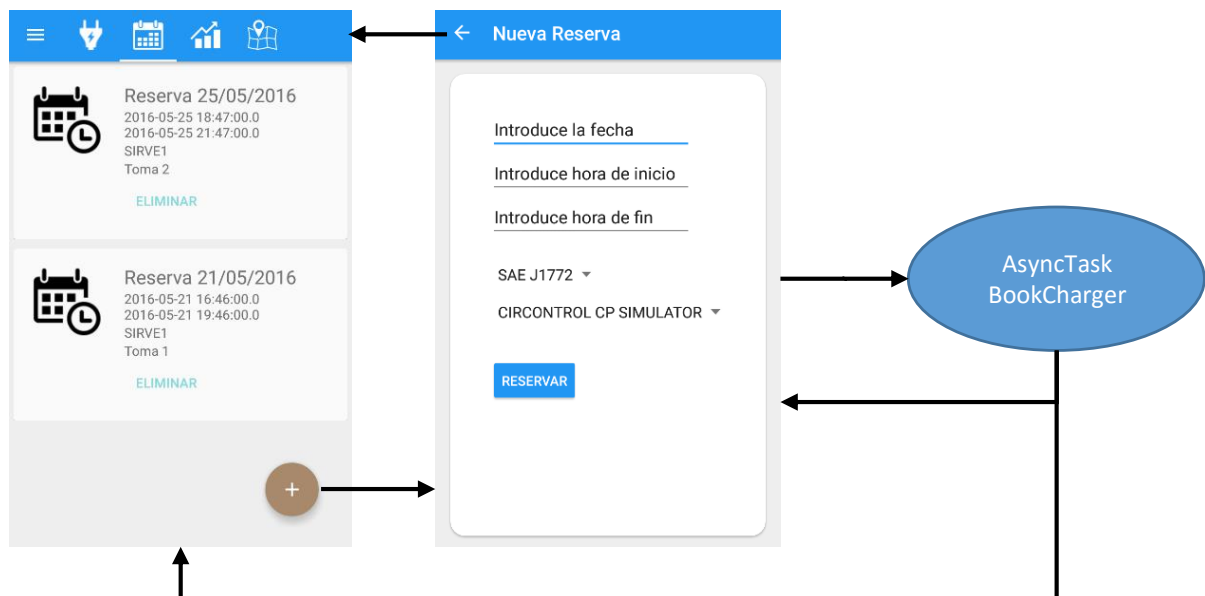


Figura 25. Mapa de navegación: nueva reserva.

Anexo C

Configuración de WS-Security

En este anexo se adjuntan los distintos archivos que han sido necesarios para la especificación de las políticas, además de ejemplos del intercambio de mensajes.

1.1 Seguridad a nivel de mensaje: algoritmo de doble cifrado

En la figura 11 se puede observar la primera parte de la política añadida en el módulo de Rampart y que deben cumplir todos los mensajes SOAP para que sean aceptados por el servidor web. En esta especificación se establece que tiene que aparecer siempre un Token del tipo <sp:509Token>, que transporta la información del certificado público en todo mensaje desde el iniciador (cliente) al servidor. En la dirección contraria, del servidor al cliente, no hay que añadir nunca este Token. Por otro lado, añade que el algoritmo utilizado para el cifrado es el <sp:Basic128/> y que el orden de aparición de los distintos elementos es obligatorio (<sp:Strict />).

```
<wsp:Policy xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
  xmlns:wsu="../oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <wsp:ExactlyOne>
    <wsp:All>
      <sp:AsymmetricBinding
        <wsp:Policy>
          <sp:InitiatorToken>
            <wsp:Policy>
              <sp:X509Token sp:IncludeToken="../IncludeToken/AlwaysToRecipient">
                <wsp:Policy>
                  <sp:WssX509V3Token10 />
                </wsp:Policy>
              </sp:X509Token>
            </wsp:Policy>
          </sp:InitiatorToken>
          <sp:RecipientToken>
            <wsp:Policy>
              <sp:X509Token sp:IncludeToken="../IncludeToken/Never">
                <wsp:Policy>
                  <sp:WssX509V3Token10 />
                </wsp:Policy>
              </sp:X509Token>
            </wsp:Policy>
          </sp:RecipientToken>
          <sp:AlgorithmSuite>
            <wsp:Policy>
              <sp:Basic128 />
            </wsp:Policy>
          </sp:AlgorithmSuite>
          <sp:Layout>
            <wsp:Policy>
              <sp:Strict />
            </wsp:Policy>
          </sp:Layout>
          <sp:IncludeTimestamp />
          <sp:OnlySignEntireHeadersAndBody />
        </wsp:Policy>
      </sp:AsymmetricBinding>
    </wsp:All>
  </wsp:ExactlyOne>
</wsp:Policy>
```

Figura 26. Política del algoritmo de doble cifrado (I)

La segunda parte de la política se puede ver en la figura 12. En esta especificación se detallan las partes del mensaje que van a ir encriptadas y firmadas con las etiquetas <sp:SignedParts> y <sp:EncryptedParts>, que en ambos casos sera el cuerpo (Body) del mensaje. Por último, aparece la configuración de Rampart, que aporta la información relacionada con la clase java que gestiona la llegada y el envío de los distintos Tokens, encriptaciones y firmas y, además, las claves utilizadas tanto para cifrar como para descifrar los mensajes.

```

        <sp:SignedParts xmlns:sp="../securitypolicy">
            <sp:Body />
        </sp:SignedParts>
        <sp:EncryptedParts xmlns:sp="../securitypolicy">
            <sp:Body />
        </sp:EncryptedParts>
        <ramp:RampartConfig xmlns:ramp="../rampart/policy">
            <ramp:user>server</ramp:user>
            <ramp:encryptionUser>client</ramp:encryptionUser>
            <ramp:passwordCallbackClass>
                web.service.servidor.SOAP.PWCBHandler
            </ramp:passwordCallbackClass>
            <ramp:encryptionCypto>
                <ramp:crypto provider="org.apache.ws.security.components.crypto.Merlin">
                    <ramp:property
                        name=".ws.security.crypto.merlin.keystore.type">JKS</ramp:property>
                    <ramp:property
                        name="org.apache.ws.security.crypto.merlin.file">server.jks</ramp:property>
                    <ramp:property
                        name=".security.crypto.merlin.keystore.password">password</ramp:property>
                </ramp:crypto>
            </ramp:encryptionCypto>
            <ramp:signatureCrypto>
                <ramp:crypto provider="org.apache.ws.security.components.crypto.Merlin">
                    <ramp:property
                        name=".ws.security.crypto.merlin.keystore.type">JKS</ramp:property>
                    <ramp:property
                        name="org.apache.ws.security.crypto.merlin.file">
server.jks</ramp:property>
                    <ramp:property
                        name=".security.crypto.merlin.keystore.password">password</ramp:property>
                </ramp:crypto>
            </ramp:signatureCrypto>
        </ramp:RampartConfig>
    </wsp:All>
</wsp:ExactlyOne>
</wsp:Policy>

```

Figura 27. Política del algoritmo de doble cifrado (II)

A continuación, en la figura 13, podemos ver un ejemplo de mensaje SOAP que cumple con la política explicada previamente. La cabecera de seguridad cuenta con tres elementos principales: <xenc:EncryptedKey>, que incluye la información de la clave secreta encriptada, la identificación de la clave utilizada para encriptarla (pública del servidor) y también el algoritmo de cifrado que se ha utilizado (rsa-oaep-mgf1p). Por otro lado, se encuentra <wsse:BinarySecurityToken>, que contiene el Token con el certificado público del cliente, y <ds:Signature>, que representa la firma digital. Finalmente, aparece el cuerpo del mensaje encriptado bajo el elemento <xenc:EncryptedData>, donde se especifica el algoritmo de cifrado utilizado.

```

<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  xmlns:soap1="http://SOAP.servidor.service.web">
  <soap:Header>
    <wsse:Security soap:mustUnderstand="true">
      <xenc:ReferenceList
        xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
        <xenc:DataReference URI="#ED-123" />
      </xenc:ReferenceList>
      <wsse:BinarySecurityToken
        EncodingType="..#Base64Binary"
        ValueType="..#X509PKIPathv1"
        wsu:Id="X509-123">MIICbDCCAmgwggHRoAMCAQICB.....
      </wsse:BinarySecurityToken>
      <ds:Signature xmlns:ds="..#xmldsig#"
        Id="SIG-BAB2C324554217A8F9145623681302489">
        <ds:SignedInfo>
          <ds:CanonicalizationMethod
            Algorithm="..#xml-exc-c14n#"
            <ec:InclusiveNamespaces xmlns:ec="..#xml-
              exc-c14n#"
              PrefixList="soap soap1 soapenv" />
          </ds:CanonicalizationMethod>
          <ds:SignatureMethod Algorithm="..#rsa-sha1" />
          <ds:Reference URI="#id-123">
            <ds:Transform
              Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"
              <ec:InclusiveNamespaces
                xmlns:ec="..#xml-exc-c14n#" PrefixList="soap1 soapenv" />
            </ds:Transform>
            </ds:Transforms>
            <ds:DigestMethod Algorithm="..#sha1" />
            <ds:DigestValue>BDpKm0U2WWrCtp2WY9Fm1si82RM=</ds:DigestValue>
          </ds:Reference>
          </ds:SignedInfo>
        </ds:SignatureValue>IZJOi1CxtuL9jQHv4cm4SOLuq1Oc.....</ds:SignatureValue>
        <ds:KeyInfo Id="KI-123">
          <wsse:SecurityTokenReference
            wsse11:TokenType="..#X509PKIPathv1"
            wsu:Id="STR-123">
              <wsse:Reference URI="#X509-123"
                ValueType="..#X509PKIPathv1" />
            </wsse:SecurityTokenReference>
          </ds:KeyInfo>
        </ds:Signature>
        <wsu:Timestamp wsu:Id="TS-BAB2C324554217A8F9145623681302085">
          <wsu:Created>2016-02-23T14:13:33.020Z</wsu:Created>
          <wsu:Expires>2016-02-23T22:33:33.020Z</wsu:Expires>
        </wsu:Timestamp>
      </wsse:Security>
    </soap:Header>
    <soap:Body wsu:Id="id-123">
      <xenc:EncryptedData xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
        Id="ED-123" Type="http://www.w3.org/2001/04/xmlenc#Content">
        <xenc:EncryptionMethod
          Algorithm="http://www.w3.org/2001/04/xmlenc#tripledes-cbc" />
        <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#" />
        <xenc:CipherData>
          <xenc:CipherValue>
            +xw0/SXvu6fMtIM2qdLJpx63O6sdBZiEeeEs0dwGZ.....
          </xenc:CipherValue>
        </xenc:CipherData>
      </xenc:EncryptedData>
    </soap:Body>
  </soap:Envelope>

```

Figura 28. Ejemplo de mensaje SOAP (II).

1.2 Seguridad a nivel de transporte

La política relativa a la seguridad por https la encontramos en la figura 14. En esta especificación es necesario detallar que se va a utilizar seguridad en la capa de transporte mediante el atributo `<sp:Transport Binding>` y el Token de transporte necesario para la comunicación segura `<sp:HttpsToken>`. Por otro lado, se ha añadido un Token de soporte para asegurar la autenticación de los usuarios y que consistirá en un `<sp:UsernameToken>`.

```
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:ns="http://SOAP.servidor.service.web"
xmlns:ax21="http://ComplexTypes.SOAP.servidor.service.web/xsd"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
targetNamespace="http://SOAP.servidor.service.web">
  <wsp:Policy xmlns:wsp="http://www.w3.org/ns/ws-policy"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="UTOverTransport">
    <wsp:ExactlyOne>
      <wsp:All>
        <sp:TransportBinding
xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy">
          <wsp:Policy>
            <sp:TransportToken>
              <wsp:Policy>
                <sp:HttpsToken RequireClientCertificate="false"/>
              </wsp:Policy>
            </sp:TransportToken>
            <sp:AlgorithmSuite>
              <wsp:Policy>
                <sp:Basic256/>
              </wsp:Policy>
            </sp:AlgorithmSuite>
            <sp:Layout>
              <wsp:Policy>
                <sp:Lax/>
              </wsp:Policy>
            </sp:Layout>
          </wsp:Policy>
        </sp:TransportBinding>
        <sp:SupportingTokens
xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy">
          <wsp:Policy>
            <sp:UsernameToken
sp:IncludeToken="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy/IncludeToken/AlwaysToRecipient"/>
          </wsp:Policy>
        </sp:SupportingTokens>
      </wsp:All>
    </wsp:ExactlyOne>
  </wsp:Policy>
```

Figura 29. Política de la seguridad a nivel de transporte.

Por último, en la figura 15 se adjunta un ejemplo de un mensaje SOAP mandado desde la aplicación móvil invocando al servicio web *Authenticate*. Como especifica la política, es obligatoria la aparición de un UsernameToken de tipo #PasswordText que, como se puede ver, contiene la contraseña encriptada.

```
<soapenv:Envelope xmlns:soap="http://SOAP.servidor.service.web"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <wsse:Security
      xmlns:wsse="../oasis-200401-wss-wssecurity-secext-1.0.xsd"
      xmlns:wsu="../oasis-200401-wss-wssecurity-utility-1.0.xsd">
      <wsse:UsernameToken wsu:Id="UsernameToken63DF73F434660">
        <wsse:Username>circe</wsse:Username>
        <wsse:Password
          Type="../oasis-200401-wss-username-token-profile-1.0#PasswordText">
          CQS0RRyY2wN13xgbD+QJnfGVhW01c9gIhmh8ngjytB+m2b9
/gt5we+M+5FwrsTyyy6hBBCVTKnzrbGA+ppRMMuDUVmL585
R4/PKK0KSywRhImlyI2s0KQANRDdWuWIZkXiZ85ldQSxc9l
        </wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soapenv:Header>
  <soapenv:Body>
    <soap:authenticate>
      <soap:username>circe</soap:username>
      <soap:password>circe</soap:password>
    </soap:authenticate>
  </soapenv:Body>
</soapenv:Envelope>
```

Figura 30. Ejemplo mensaje SOAP (II).

Anexo D

Ciclo de vida de la aplicación

En este anexo se adjunta una tabla (figura 16) que detalla la gestión de la persistencia y las variables globales durante las distintas etapas del ciclo de vida de la aplicación.

	ActiveCharge	Other Activities
OnStart	<p>Si hay datos de login en Shared Preferences: Si no están cargados, carga los datos de usuario en las variables globales.</p> <p>Si hay datos de carga activa: Si no están cargados, carga los datos de la carga activa en variables globales.</p> <p>Si no hay datos de login: Inicia la pantalla de Login.</p> <p>Iniciar las peticiones de medidas.</p>	<p>Si hay datos de login en Shared Preferences: Si no están cargados, carga los datos de usuario en las variables globales.</p> <p>Si hay datos de carga activa: Carga los datos de la carga activa en variables globales. Inicia la actividad de carga activa.</p> <p>Si no hay datos de login: Inicia la pantalla de Login.</p>
OnPause	<p>Parar las peticiones de medidas.</p> <p>Eliminar los datos de usuario de las variables globales.</p>	<p>Eliminar los datos de usuario de las variables globales.</p>
OnResume	<p>Seguir mandando peticiones.</p>	
OnDestroy	<p>Eliminar los datos de usuario de las variables globales.</p> <p>Parar las peticiones de medidas.</p>	<p>Eliminar los datos de usuario de las variables globales.</p>

Figura 31. Ciclo de vida de la aplicación

Anexo E

Descripción y resultados de las pruebas y test

1. Test unitarios

A continuación, en la figura 17, se muestra la tabla de test de unidad realizados para validar el correcto funcionamiento de la aplicación y poder desarrollar nuevo software de una manera ágil, asegurando que sigue funcionando de igual manera que en antiguas versiones.

Activity	Condiciones	Acción	Comportamiento deseado
Login	No rellenar campos de usuario / contraseña	Botón login	Error: Introduce usuario/contraseña
	Contraseña no adecuada (longitud)		Error: Introduce contraseña válida
	Usuario / contraseña no corresponden		Error: Error de autenticación
	Login correcto		Iniciar Main Activity
	Fallo de conexión		Mensaje de error: fallo de conexión.
Main		Deslizar ViewPager	Pasar fragmentos
		Botón N.Drawer	Desplegar Navigation Drawer
Main (N. Drawer)		Botón info. personal	Iniciar Personal Info Activity
		Botón ayuda	Iniciar Help Activity
Main (Scan)	QR con codificación incorrecta	Escaneo de QR	Mensaje de error: código incorrecto
	QR con codificación correcta		Iniciar Charge Info Activity
	QR con codificación correcta, fallo conexión		Mensaje de error: fallo de conexión
Charge Info Activity	Cargador disponible y autorizado	Botón iniciar carga	Iniciar Active Charge Activity

	Cargador no autorizado	Botón iniciar carga	Mensaje de error: cargador no autorizado
	Cargador no disponible	Botón iniciar carga	Mensaje de error: cargador no disponible
		Botón atrás	Iniciar Main Activity
Active Charge Activity	Petición datos de la carga, fallo de conexión.		Mensaje de error: fallo de conexión
	Petición datos de la carga		Actualización pantalla
Main (Bookings)		Botón eliminar reserva	Desaparece reserva seleccionada
	Fallo conexión, error al eliminar	Botón eliminar reserva	Alerta de que no se ha podido eliminar
		Botón añadir reserva	Inicia pantalla de nueva reserva
Select Booking Activity	Reserva correcta	Botón enviar reserva.	Reserva realizada correctamente. Vuelve a Activity principal
	Reserva incorrecta	Botón enviar reserva.	Muestra “error en la selección de fechas”
	Slot ocupado	Botón enviar reserva.	Muestra “Cargador ocupado”
Main (Statistics)			Se carga el gráfico de recargas
Main (Map)		Botón distancias	Iniciar NearChargers Activity
		Cambiar localización	Respuesta del mapa
	Servicios de localización desactivados	Botón distancias	Error conexión: necesita activar los servicios de localización
		Cambiar Discrete Seek Bar	Produce cambio OnProgressChanged

Figura 32. Tabla de test unitarios

2. Test de integración

Una vez que los test unitarios son correctos, las pruebas de integración validan el funcionamiento de la aplicación en conjunto con otros sistemas, en este caso los servicios web. El alcance de estos test abarca todos los requisitos de la aplicación y se han probado uno a uno individualmente, documentando su resultado en la siguiente tabla (Figura 18).

Caso de uso	Acción	Comportamiento deseado	Resultado
Login	Introducir los datos del usuario	Cargar datos de la base de datos e iniciar el menú principal	Ok
Analizar QR Cargador	Escanear código QR de una toma	Iniciar la actividad de recarga y ver la información del cargador y su estado y permisos	Ok
Iniciar una carga	Enchufar el coche y pulsar el botón de iniciar carga	La carga da comienzo y se inicia la pantalla de recarga activa	Ok
Obtener datos de la carga	---	En intervalos de 14 segundos, se actualiza el valor de la energía total recargada	Ok
Parar la carga	Pulsar el botón de parar carga	Finaliza la carga y se muestra un resumen de ésta	Ok
Consultar información personal	Deslizar ViewPager	Pasar fragmentos	Ok
Consultar estadísticas	Ir a la pantalla de estadísticas	Se muestra el gráfico-resumen y una lista de las últimas recargas	Ok
Consultar mapa de cargadores	Ir a la pantalla de mapa de cargadores	Se muestra el mapa con los cargadores disponibles para el usuario	Ok
Filtrar por cargadores cercanos	Seleccionar un rango de kilómetros y presionar el botón de filtrar	Inicia la pantalla de cargadores cercanos y muestra la información de los que se encuentren a una distancia menor o igual al radio elegido	Ok
Consultar reservas	Ir a pantalla de reservas	Se muestra la lista de reservas realizadas por el usuario	Ok
Eliminar una reserva	Pulsar el botón eliminar de una reserva	La reserva desaparece de la lista previa notificación correcta	Ok
Añadir una nueva reserva	Pulsar el botón de añadir nueva reserva. Rellenar los datos de la reserva y enviar.	Refresca la lista de reservas y se visualiza en la pantalla.	Ok

Figura 33. Test de integración

3. Test de usabilidad

A continuación se muestra la ficha de la prueba de usabilidad que fue pasada a los cuatro usuarios seleccionados. El moderador iba leyendo las instrucciones de la ficha, supervisando directamente e interpretando los movimientos del sujeto para evaluar el entendimiento y la rapidez, en una escala de 0-5, con la que llevaba a cabo la tarea. Una vez que el sujeto había realizado la acción, se le preguntaba si estaba satisfecho con el resultado o si mejoraría de alguna forma esa parte de la aplicación. Toda esta información se iba anotando en una hoja Excel como las que se pueden ver más adelante y que recogen todos los test que se realizaron.

Pantalla de Login

- Introduce los datos (erróneos) ¿Encuentras el error bien notificado?
- Introduce los datos (correctos).

Menú principal

- Desplázate hasta la pantalla del mapa de cargadores.

Mapa de cargadores

- Navega por el mapa
- Selecciona radio 10 Km ver cargadores (localización desactivad) ¿Qué crees que ocurre? ¿Está el GPS activo?
- Ahora sí, selecciona el rango ¿Información suficiente?
- Desplázate a la pantalla de cargas

Pantalla de cargas

- Desplázate deslizando cargas
- ¿Es útil la información del gráfico? ¿Te gustaría visualizar otro tipo de datos?
- Mira la quinta carga en la lista. ¿Entiendes la información presentada?
- Ve a la pantalla de reservas

Pantalla de reserva

- Desplázate a la pantalla de nueva reserva.
- Añade una reserva con la hora de inicio posterior a la final.
- Añade una reserva (correcta).
- ¿Puedes visualizar la reserva realizada?
- Elimina la reserva.
- Desplázate a la pantalla de gestión de recargas.

Pantalla QR

- Escanea el código QR (erróneo) ¿Eres consciente de qué ocurre?
- Escanea el código QR (correcto).
- Información cargador
- ¿Sabes interpretar la información del cargador?
- Inicia el proceso de recarga. ¿Mensaje de espera claro?

Monitorizar carga

- ¿Consideras suficiente la información proporcionada? ¿El intervalo de peticiones lo ves adecuado?
- Manda parar la recarga.
- ¿Estás satisfecho con la notificación de parada y el resumen de la recarga?
- Vuelve al menú principal.

Sujeto: Andreas Muñoz Zuara.		0 (muy mal) 1 (mal) 2 (normal) 3 (bien) 4 (perfecto)					Comentarios
Usabilidad	Entendido	Rapidez	Esperado	Satisfacción			
Pantalla de Login	El usuario ve claros los campos que tiene que rellenar.	5	5	5	3		
Menú principal	Es intuitivo ver qué campo está seleccionado.	5	5	5	3	Letras más grandes en el error	
	En caso de introducir la contraseña errónea, se notifica correctamente.	5	5	5	5		
	En caso de introducir los datos correctos, se traslada a la siguiente pantalla.	5	5	5	5		
Pantalla mapa	El usuario sabe navegar por las distintas pantallas del menú.	3	5	5	5	no se ve claro por si botones	
	Es capaz de desplegar el menú lateral con el botón.	4	5	5	5		
	Es capaz de desplegar el menú lateral deslizando.						
Pantalla cargas	El usuario sabe navegar por el mapa buscando cargadores.	5	5	5	5	activar gps no claro	
	El usuario es notificado cuando no tiene el GPS activo y sabe lo que tiene que hacer.	1	1	1	1	ratio no se entiende bien	
	La selección del rango de kilómetros es clara.	2	4	3	2	Pulsar en el cargador para reservar	
Pantalla reservas	La información de los cargadores cercanos es representativa.	5	5	5	4		
	El diagrama presenta información relevante para el usuario y es entendible.	3	5	2	2	Especificar tipo de enchufe	
	El usuario puede desplazar el menú y ver todas las cargas.	3	5	5	5	gráfica más detallada	
Pantalla nueva reserva	La información de las cargas es útil para el usuario	5	5	5	4		
	El usuario tiene clara la visualización de las reservas.	5	5	5	5		
	El usuario puede eliminar una reserva y ver como desaparece.	5	5	2	3	Modificar reservas	
Navigation Drawer	El usuario sabe ir a la pantalla de nueva reserva.	5	5	5	5		
	Es sencillo seleccionar la fechas y las horas de la nueva reserva.	5	5	5	5		
	El desplegable presenta la información requerida por el usuario para reservar.	5	5	5	3		
Pantalla QR	Al realizar una reserva errónea, el sistema notifica al usuario correctamente.	5	5	5	5		
	Al realizar una reserva correcta, el usuario puede ver la nueva reserva registrada..	4	5	2	2		
	El usuario puede acceder a su información personal.	5	5	5	5	una vez capturado, no estar siempre capturando	
Pantalla info	El usuario puede acceder a la ayuda e información.	5	5	5	4	si no es correcto	
	El usuario sabe cerrar sesión y volver a la pantalla de login.	5	5	5	5		
	El usuario enfoca el código QR erróneo y es consciente de lo que ha ocurrido.	5	5	4	4		
Pantalla carga activa	El usuario enfoca el código QR correcto y pasa a la siguiente pantalla.	5	5	5	5		
	El usuario puede interpretar la información presentada.	4	5	5	5		
	El usuario sabe como iniciar una nueva carga.	5	5	5	5		
Pantalla carga activa	El usuario recibe una notificación en caso de no ser aceptado el inicio remoto de la carga.	5	5	5	5		
	El usuario pasa a la carga activa y puede ver el progreso.	5	5	5	5		
	Si la carga es finalizada por el sistema, el usuario es notificado de la terminación.	4	5	4	4		
Comentarios	El usuario puede acabar una carga y ver su resumen.	5	5	5	5		
	Los datos que aparecen en el resumen son representativos.	5	5	5	4		

Sujeto: Hans Bludswelt		0 (muy mal) 1 (mal) 2 (normal) 3 (bien) 4 (perfecto)					Comentarios
	Usabilidad	Entendido	Rapidez	Esperado	Satisfacción		
Pantalla de Login							
El usuario ve claros los campos que tiene que rellenar.		5	5	5	5		
Es intuitivo ver qué campo está seleccionado.		5	5	5	5		
En caso de introducir la contraseña errónea, se notifica correctamente.		5	5	5	5		
En caso de introducir los datos correctos, se traslada a la siguiente pantalla.		5	5	5	5		
Menú principal							
El usuario sabe navegar por las distintas pantallas del menú.		5	5	5	5	tiende a pulsar boton en vez de deslizar	
Es capaz de desplegar el menú lateral con el botón .		5	5	5	5		
Es capaz de desplegar el menú lateral deslizando.		5	5	5	5		
Pantalla mapa							
El usuario sabe navegar por el mapa buscando cargadores.		3	3	3	3	Cambiar zoom al elegir rango o ir directo a pantalla cargadores	
El usuario es notificado cuando no tiene el GPS activo y sabe lo que tiene que hacer.		5	5	5	5	estado del cargador, ver si hay hueco	
La selección del rango de kilómetros es clara.		4	5	4	4		
La información de los cargadores cercanos es representativa.							
Pantalla cargas							
El diagrama presenta información relevante para el usuario y es entendible.		3	3	3	3		
El usuario puede desplazar el menú y ver todas las cargas.		5	5	5	5		
La información de las cargas es útil para el usuario		4	5	4	4		
Pantalla reservas							
El usuario tiene clara la visualización de las reservas.		5	5	5	5	errores concretos en los campos de reserva.	
El usuario puede eliminar una reserva y ver como desaparece.		5	5	5	5		
El usuario sabe ir a la pantalla de nueva reserva.		5	5	5	5		
Pantalla nueva reserva							
Es sencillo seleccionar la fechas y las horas de la nueva reserva.		5	5	5	4		
El desplegable presenta la información requerida por el usuario para reservar.		5	5	5	5		
Al realizar una reserva errónea, el sistema notifica al usuario correctamente.		5	5	5	5		
Al realizar una reserva correcta, el usuario puede ver la nueva reserva registrada..		4	4	3	3		
Navigation Drawer							
El usuario puede acceder a su información personal.		5	5	5	5		
El usuario puede acceder a la ayuda e información.		5	5	5	5		
El usuario sabe cerrar sesión y volver a la pantalla de login.		5	5	5	5		
Pantalla QR							
El usuario enfoca el código QR erróneo y es consciente de lo que ha ocurrido.		2	4	2	2		
El usuario enfoca el código QR correcto y pasa a la siguiente pantalla.		5	5	5	5		
Pantalla información cargador							
El usuario puede interpretar la información presentada.		5	5	5	5		
El usuario sabe como iniciar una nueva carga.		5	5	5	5		
El usuario recibe una notificación en caso de no ser aceptado el inicio remoto de la carga.		5	5	5	5		
Pantalla carga activa							
El usuario pasa a la carga activa y puede ver el progreso.		5	5	5	5		
Si la carga es finalizada por el sistema, el usuario es notificado de la terminación.		5	5	5	3		
El usuario puede acabar una carga y ver su resumen.		5	5	5	5	Recargo monteario en el proceso de recarga	
Los datos que aparecen en el resumen son representativos.		5	5	5	5		

Sujeto: José María Gimeno Conejos		0 (muy mal) 1 (mal) 2 (normal) 3 (bien) 4 (perfecto)					Comentarios
Usabilidad		Entendido	Rapidez	Esperado	Satisfacción		
Pantalla de Login							
El usuario ve claros los campos que tiene que rellenar.		5	5	5	5		
Es intuitivo ver qué campo está seleccionado.		5	5	5	5		
En caso de introducir la contraseña errónea, se notifica correctamente.		5	5	5	5		
En caso de introducir los datos correctos, se traslada a la siguiente pantalla.		5	5	5	5		
Menú principal							
El usuario sabe navegar por las distintas pantallas del menú.		5	5	5	5		
Es capaz de desplegar el menú lateral con el botón .		5	5	5	5		
Es capaz de desplegar el menú lateral deslizando.		3	3	3	3		
Pantalla mapa							
El usuario sabe navegar por el mapa buscando cargadores.		5	5	5	5		Km rango reservar desde el mapa
El usuario es notificado cuando no tiene el GPS activo y sabe lo que tiene que hacer.		4	4	4	4		Tipo carga de cada estación
La selección del rango de kilómetros es clara.		2	2	2	2		
La información de los cargadores cercanos es representativa.		5	5	5	5		
Pantalla cargas							
El diagrama presenta información relevante para el usuario y es entendible.		5	5	5	5		seleccionar estadísticas por mes
El usuario puede desplazar el menú y ver todas las cargas.		3	3	2	2		
La información de las cargas es útil para el usuario		5	4	4	4		
Pantalla reservas							
El usuario tiene clara la visualización de las reservas.		5	5	5	5		Notificar que se acerca una reserva
El usuario puede eliminar una reserva y ver como desaparece.		5	5	5	5		
El usuario sabe ir a la pantalla de nueva reserva.		5	5	5	5		
Pantalla nueva reserva							
Es sencillo seleccionar la fechas y las horas de la nueva reserva.		5	5	5	5		
El desplegable presenta la información requerida por el usuario para reservar.		5	5	5	5		
Al realizar una reserva errónea, el sistema notifica al usuario correctamente.		5	5	4	4		
Al realizar una reserva correcta, el usuario puede ver la nueva reserva registrada..							
Navigation Drawer							
El usuario puede acceder a su información personal.		5	5	5	5		
El usuario puede acceder a la ayuda e información.		5	5	5	5		
El usuario sabe cerrar sesión y volver a la pantalla de login.		5	5	5	5		
Pantalla QR							
El usuario enfoca el código QR erróneo y es consciente de lo que ha ocurrido.		4	4	4	4		
El usuario enfoca el código QR correcto y pasa a la siguiente pantalla.		5	5	5	5		
Pantalla información cargador							
El usuario puede interpretar la información presentada.		5	5	5	5		
El usuario sabe como iniciar una nueva carga.		5	5	5	5		
El usuario recibe una notificación en caso de no ser aceptado el inicio remoto de la carga.		5	5	5	5		
Pantalla carga activa							
El usuario pasa a la carga activa y puede ver el progreso.		5	5	5	5		Alerta notificación de parada, personalizable en ajustes.
Si la carga es finalizada por el sistema, el usuario es notificado de la terminación.		5	5	5	5		Opción de datos avanzados pero no importantes
El usuario puede acabar una carga y ver su resumen.		5	5	5	5		
Los datos que aparecen en el resumen son representativos.		5	5	5	4		

Sujeto: José Francisco Sanz Osorio		0 (muy mal) 1 (mal) 2 (normal) 3 (bien) 4 (perfecto)				Comentarios
Usabilidad		Entendido	Rapidez	Esperado	Satisfacción	
Pantalla de Login						
El usuario ve claros los campos que tiene que rellenar.		5	5	5	5	
Es intuitivo ver qué campo está seleccionado.		5	5	5	5	
En caso de introducir la contraseña errónea, se notifica correctamente.		4	5	5	5	
En caso de introducir los datos correctos, se traslada a la siguiente pantalla.		5	5	5	5	
Menú principal						
El usuario sabe navegar por las distintas pantallas del menú.		5	5	5	5	
Es capaz de desplegar el menú lateral con el botón.		4	5	5	5	
Es capaz de desplegar el menú lateral deslizando.		3	3	2	4	
Pantalla mapa						
El usuario sabe navegar por el mapa buscando cargadores.		5	4	5	5	Cambiar zoom al elegir rango
El usuario es notificado cuando no tiene el GP'S activo y sabe lo que tiene que hacer.		4	5	5	5	Planificar ruta y elegir las estaciones de recarga más óptimas.
La selección del rango de kilómetros es clara.		3	4	4	4	Permitir reservar al pinchar en un cargador
La información de los cargadores cercanos es representativa.		3	5	3	3	
Pantalla cargas						
El diagrama presenta información relevante para el usuario y es entendible.		3	3	3	2	Filtrar cargas por mes al pulsar en las columnas del gráfico.
El usuario puede desplazar el menú y ver todas las cargas.		5	5	5	5	
La información de las cargas es útil para el usuario		4	5	4	3	
Pantalla reservas						
El usuario tiene clara la visualización de las reservas.		3	5	5	4	
El usuario puede eliminar una reserva y ver como desaparece.		5	5	5	5	Poner rótulos de tipo de toma y cargador.
El usuario sabe ir a la pantalla de nueva reserva.		5	5	5	5	
Pantalla nueva reserva						
Es sencillo seleccionar la fechas y las horas de la nueva reserva.		5	4	5	5	
El desplegable presenta la información requerida por el usuario para reservar.		4	5	5	3	
Al realizar una reserva errónea, el sistema notifica al usuario correctamente.		5	5	5	5	
Al realizar una reserva correcta, el usuario puede ver la nueva reserva registrada..		4	4	4	4	
Navigation Drawer						
El usuario puede acceder a su información personal.		5	5	5	5	
El usuario puede acceder a la ayuda e información.		5	5	5	5	
El usuario sabe cerrar sesión y volver a la pantalla de login.		5	5	5	5	
Pantalla QR						
El usuario enfoca el código QR erróneo y es consciente de lo que ha ocurrido.		5	5	5	5	
El usuario enfoca el código QR correcto y pasa a la siguiente pantalla.		5	5	5	5	
Pantalla información cargador						
El usuario puede interpretar la información presentada.		5	5	4	4	
El usuario sabe como iniciar una nueva carga.		5	5	5	5	
El usuario recibe una notificación en caso de no ser aceptado el inicio remoto de la carga.		5	5	5	5	
Pantalla carga activa						
El usuario pasa a la carga activa y puede ver el progreso.		5	5	5	5	
Si la carga es finalizada por el sistema, el usuario es notificado de la terminación.		5	5	5	3	
El usuario puede acabar una carga y ver su resumen.		5	5	5	5	
Los datos que aparecen en el resumen son representativos.		5	5	4	4	